REPORT NO.

INFO-C

71158

NUCLEAR ROCKET OPERATIONS
SACRAMENTO PLANT

REPORT NO. RN-S-0427

TO
AEC-NASA SPACE NUCLEAR PROPULSION OFFICE

XE DATA ANALYSIS PROGRAM

NERVA PROGRAM

CONTRACT SNP-1

SEPTEMBER 1967

MASTER

100-100000-1

DISCLAIMER

This report was prepared as an account of work sponsored by an agency of the United States Government. Neither the United States Government nor any agency Thereof, nor any of their employees, makes any warranty, express or implied, or assumes any legal liability or responsibility for the accuracy, completeness, or usefulness of any information, apparatus, product, or process disclosed, or represents that its use would not infringe privately owned rights. Reference herein to any specific commercial product, process, or service by trade name, trademark, manufacturer, or otherwise does not necessarily constitute or imply its endorsement, recommendation, or favoring by the United States Government or any agency thereof. The views and opinions of authors expressed herein do not necessarily state or reflect those of the United States Government or any agency thereof.

DISCLAIMER

Portions of this document may be illegible in electronic image products. Images are produced from the best available original document.

REPORT NO. RN-S-0427

XE DATA ANALYSIS PROGRAM

NERVA PROGRAM

CONTRACT SNP-1

NUCLEAR ROCKET OPERATIONS

SEPTEMBER 1967

NOTICE

This report was prepared as an account of work sponsored by the United States Government. Neither the United States nor the United States Atomic Energy Commission, nor any of their employees, nor any of their contractors, subcontractors, or their employees, makes any warranty express or implied, or assumes any legal liability or responsibility for the accuracy, completeness or usefulness of any information, apparatus, product or process disclosed, or represents that its use would not infringe privately owned rights.

CLASSIFICATION CATEGORY

UNCLASSIFIED

N. H. Slonim
CLASSIFYING OFFICER

Sept. 25, 1967
DATE

AEROJET-GENERAL CORPORATION
A SUBSIDIARY OF THE GENERAL TIRE & RUBBER COMPANY

BLANK

REPORT NO. RN-S-0427

XE DATA ANALYSIS PROGRAM

N. H. Slonim
G. G. Strucel

A handwritten signature in black ink, appearing to read 'C. M. Rice', is positioned above the printed name.

C. M. Rice
Program Manager
Nuclear Rocket Operations

BLANK

ABSTRACT

This report describes the first phase of a narrow-band data analysis computer program developed by Aerojet-General Corporation for handling test data in the NERVA program. The program is written in FORTRAN IV language for processing on the IBM 360-65 computers. Capabilities of the program include the preparation of multiple plots of test data, the performance of engineering calculations such as for flow rate and thermal power, the smoothing of data to minimize the noise from the data for purposes of evaluation, the preparation of steady-state summaries of selected time slices for component and system evaluation, the listing of measurement anomalies to identify missing or off-scale data and to pinpoint bad data recordings, and the listing of events for the evaluation of test-system performance.

Descriptions covering input to, and output from, the program are given in detail.

BLANK

CONTENTS

	<u>Page</u>
I. Introduction	1
II. Summary	3
III. Technical Discussion	5
A. Operation of the Data Program	5
1. General	5
2. Preparation for a Run	6
3. Display of Results	7
B. Features in Phase-1 Version of Program	11
1. Program Output	11
a. Data Plots	11
b. Steady-State Listings	16
c. Data Anomalies	16
d. Events Listing	18
2. Data Input	19
a. Tape-Read Routines	19
b. Check of Channels Used in Pretest Planning	21
3. Data Calculations	21
a. Raw-Data Smoothing	21
b. Channel Averaging	27
c. Flow Calculations	27
d. Special Calculations	32
4. Retention of Data	34

APPENDIXES

A. Card Input Format	37
B. Flow Diagram and FORTRAN Source-Program Listing, X-Engine Data Program	67
C. ETS-1 Engineering-Unit Tape Format	169
D. Conversion of Modified CDC 3200 Floating-Point Numbers to the Equivalent IBM 360 Floating-Point Representation	179

TABLE

<u>No.</u>	<u>Title</u>	<u>Page</u>
1	Units of Analysis for X-Engine Testing	17

FIGURES

<u>No.</u>	<u>Title</u>	<u>Page</u>
1	Sample Plot, Duct Water Pressure at Inlet and Outlet	12
2	Sample Plot, Operating Conditions in Primary Chamber and at Primary Nozzle Exit	13
3	Smoothing Routine, Characteristic Sine-Wave Input	23
4	CFDTS Data, Run 19	24
5	NRX/EST EP IIC Data, Turbopump Speed	25
6	NRX/EST EP IIC Data, Reflector Inlet Plenum Pressure	26
7	Flow Calculations	28

I. INTRODUCTION

Aerojet-General Corporation is developing an automatic process to handle test data for the NERVA engine test program. The objective of the data program is to perform the routine data reduction so that engineering personnel can concentrate on the analysis of data. The program utilizes the capabilities of the third-generation IBM 360-65 computer. This report describes the first phase of the Aerojet-General NERVA data-analysis program (narrow band), which is currently operational and has been used with facility test data from ETS-1.

Specifically, the program performs the following functions:

- (1) Multiple plots of test data, permitting a rapid comparison of the performance of various engine parameters.
- (2) Performance of engineering calculations, such as for flow rate and thermal power, which are required for engineering system evaluation.
- (3) Data smoothing to minimize the noise from the data for evaluation.
- (4) Steady-state summaries of selected time slices for component and system evaluation.
- (5) Measurement anomalies to identify missing or off-scale data and indicate bad data channels.
- (6) Event listing for the evaluation of test-system performance.

The foregoing objectives are met by generating cathode-ray-tube multiple plots for transient conditions and a digital listing for events, measurement anomalies, and steady-state summaries. The program has been designed for rapid operation to meet the requirements of the NERVA program.

In particular, the program can complete its work within one day after the receipt of data tapes, including up to 1100 channels of measurements, from the Nevada Test Operations and can provide the information to the Systems Performance, Evaluation, and Analysis Review team (SPEAR) for their evaluation. Since the Nevada Test Operations will require one day to generate the necessary engineering-unit tapes required as input to the Aerojet-General NERVA data-analysis program, the data will be available at Nevada Test Operations in a processed form for detailed engineering evaluation two days after an experimental test is run. Also, the output of the program is of sufficient detail that it can be used as a preliminary data-analysis report on each engine test. The use of the program will permit efforts to be concentrated on system performance anomalies, upgrading of computer models, and performance evaluations to relate experimental tests, information available on the hardware, and previous system and component tests. The program format is flexible so that it can be adapted to the analysis of data generated at any point in the NERVA engine program by changing the tape-read routines. In fact, the program can be used to efficiently analyze data from any source where a large number of channels are being measured at a given time.

The program is being expanded in phases to better meet the requirements of the engineers analyzing the data and to permit concentration on engineering problems. Among the features now being added are:

- (1) Application of statistical techniques to the data for an understanding of the meaningfulness of the measurements.

- (2) Comparison between the test-data measurements and analytical programs used to predict system performance based on either pretest predictions or as-run calculations to provide the analyst with a rapid evaluation of his knowledge of the system.

II. SUMMARY

Phase 1 of the data-analysis program is operational and has been placed on production in the Computing Sciences Library at Aerojet's Sacramento Plant. The program was checked out on tapes obtained from NEPN-1 (a duct test conducted in ETS-1). The program successfully produced cathode-ray-tube multiplots of the data, digital listings of the engineering-unit data, and steady-state averages of all data over selected time periods.

Data were averaged, smoothed, and used in special calculations. A complete listing of features in Phase 1 of the data program is covered in Section B of the technical discussion. Details of program input and output are included in Appendix A. Flow diagrams and computer source program listings in FORTRAN IV language may be found in Appendix B.

Programming is under way for the second phase of the program. In this effort, emphasis is being placed on statistical analysis of steady-state holds, incorporation of instrumentation uncertainties, and inclusion of analytical programs for comparison with test results. Improving the program with the addition of new features will be a continuing effort which will be completed in phases in order to ensure that an operational version of the program is always available for test analysis.

BLANK

III. TECHNICAL DISCUSSION

A. OPERATION OF THE DATA PROGRAM

1. General

The data analysis program has been developed to be run on an IBM 360-65 computer by using engineering-unit tapes prepared by NTO. The program takes advantage of (1) a computer with a large core and disk storage that allows a rapid conversion of the data into plot format, (2) a cathode-ray-tube plotter that can quickly handle the large volume of plots required from an engine test, and (3) the operational engine systems programs and nonnuclear component programs that can be quickly used to compute the expected performance of the engine for the as-run conditions.

To eliminate interface problems, close coordination has been maintained between NRO in Sacramento and the NTO ETS-1 data group during the data program development. The Sacramento program will use the NTO engineering-unit tapes. These tapes have been corrected by NTO for channel calibration errors, and no additional calibration checks will be made by Sacramento. The NTO tape format is completely compatible with the Sacramento IBM 360-65 system, and periodic meetings are held with NTO to ensure that any contemplated changes can be handled with the data program.

During any test to be analyzed by using the data program on a priority basis (plots for SPEAR by R + 2), a Sacramento engineer will be present at NTO in the computer room to monitor the formation of the engineering-unit tapes. He will maintain a log of the tape formation to ensure that any unplanned occurrences are properly recorded so that they do not delay data processing at Sacramento. He, or an equally well-informed engineer, will accompany the tapes to Sacramento to monitor processing on the IBM 360-65.

A detailed procedure has been established to ensure that all information on the test-data tapes is properly taken into account. Input instructions for the data program will be pre-prepared and will be on tape (see Section III.A.2). Three days before each test, NTO intends to conduct a final checkout of the data system. When this test is conducted, a deck of cards will be prepared that identifies all channels to be used in the test. These cards will be brought to Sacramento and used to check the data-program input tape.

The NTO cards and the data-program input instructions are put into a routine that checks for channel discrepancies between the two. A printout is obtained that indicates channels that are present in the data program and not in the NTO card deck and those that are on the NTO card deck and not in the data program input. The data program can then be corrected with override cards in order to ensure a one-to-one matching with the channels to be used on the test.

After run day, when the NTO data tapes are received at Sacramento, the NTO card deck for the as-run test is again checked with the program input to be sure that last-minute changes in the data system are accounted for.

2. Preparation for a Run

The data program is being developed with a major emphasis on preparing data that can be easily read and interpreted by data analysts. To meet this objective, the program has been designed with a large number of input options. It is necessary to initiate planning for the data analysis well before each test is conducted. Planning for a typical test will be initiated with the issuance of the Recording and Display Requirements (RDR), the NTO document that first combines the engine and facility measurement channels to be used in a given test. This document should be available about three months before a test.

The planned instrumentation will be reviewed by the Data Analysis Section of the NRO Systems Analysis Dept., and a data analysis plan will be prepared. This plan will include all special parameters to be computed with the data, the arrangements and scales for each multiplot, and the arrangements for the steady-state summaries. All interested engineering and data-analysis groups will be contacted to ensure that all needs are being met. Two months before the test, a memorandum will be published that defines the analysis plan. This information will be sent to all interested engineering groups and to the SPEAR Director. Comments, requests for changes, and requests for additional plots will be limited to a maximum time of one month from the issue date of the memorandum, or approximately one month prior to the test.

Preparation of the program input cards will be initiated with the issuance of the memorandum. Requested changes will be handled manually until the one-month pretest deadline. At this time, the card inputs will be put on tape and frozen as the data plan for the subject test. The tape can be changed with override cards, as mentioned in Section III.A.1; but an excess of last-minute changes could lead to input errors and will be avoided as much as possible. NTO has a data plotting program that can handle a smaller volume of plots, and it should be used for most last-minute individual needs that were not previously anticipated.

3. Display of Results

A major effort has been expended to make the program output a usable package that best displays the data relative to the test objectives. All output will be combined into an 11- by 17-in. hard-cover book to which data can be added or removed. Records will be maintained on all books issued, and, if secondary data runs are made, inserts will be sent to the data-book holders. When classified data are included in a test, CRD and non-CRD books can be issued. The relative volume of classified and nonclassified channels

will be reviewed before each test, and a pretest decision will be made as to whether to issue one or two volumes. This information will be transmitted along with other information in the planning memorandum, issued two months before the test.

The data book will normally contain five sections, each tabbed and separated with a rigid divider. The content of each major section is summarized in the following paragraphs.

a. Section No. 1: Description of Special Calculations

A writeup on all special calculations performed for any given test will be included in this section. The writeup will define the data channels used to compute each new parameter and the exact mathematical formulas used. Where parameters have been generated from best-estimate routines, a complete description of the best-estimate routine will be included.

All new parameters computed by the program will be given an eight-digit identifying number similar in format to the channel data log. The new parameters will be indexed alphanumerically for ease of location.

b. Section No. 2: Summary of Anomalies

The program automatically records and prints out any missing data or anomalies it encounters during a data run. The information will be recorded in this section of the data book. In addition to the computer output, any unusual occurrences noted by the program operator that could be useful to the data analysts will be included as notes in this section.

c. Section No. 3: Plots of Data

All cathode-ray-tube plots of the data will be in this section. The plots will be arranged by units of interest, such as pump parameters, turbine parameters, and nozzle parameters. By finding the plots that would normally be used for a specific analysis all together in one area, the user can go directly to his immediate area of concern and thereby obviate a search for the plots in other sections of the data book.

A dual index is provided for the multiplots. The first index is a subject index. Each plot is given a number under the subject with which it is being displayed; e.g., the nozzle outlet temperatures might be displayed under a code number N-1. The data analyst can locate this information by looking under the subject of nozzle until he found outlet temperature. The subject plot would then be filed alphanumerically under the Code N-1. Each plot includes this code number twice, with the title and in the lower right-hand corner. The lower right-hand corner tab can be quickly observed when shuffling through the plots.

A plot such as nozzle outlet temperatures could be included twice, once under nozzles and a second time in the reflector section. In the reflector section, it would be entitled reflector inlet temperatures and would be given a different code number.

The second index is the channel index. In this case, all channels are listed alphanumerically, and the plot or plots on which they are shown are indicated. For instance, the fictitious channel T-206, which could be measuring the nozzle outlet temperature, could be shown on two plots, A-1 in the nozzle section and R-3 in the reflector section.

The indexes are prepared automatically by the computer after it has completed processing the data, so that any last-minute changes that have been added are indexed.

d. Section No. 4: Steady-State Summaries

The program will summarize all channels over any desired period. The digital listing is displayed alphanumerically by units of interest, similar to the multiplots. The subject location in the listing is defined by input, and the computer automatically makes the alphanumeric arrangement within each subject, before printing the output.

In later versions of the program, special displays of steady-state data will be shown that compare the data with predictions and computed uncertainties.

e. Section No. 5: Events Tabulation

A tabulation of all digital event data will be shown in this section. In phase one of the program, all events included in the digital data system will be tabulated alphanumerically. The starting and final positions of the event will always be shown at the time they were checked. If any event channel is changed at any time between the check periods, the change will be noted along with the time at which it occurred. These changes will be noted a maximum of 11 times. This limit has been imposed to eliminate a malfunctioning channel that is periodic and which would result in using an excessive amount of paper.

In future versions of the program, a chronological listing of the event data will also be prepared.

B. FEATURES IN PHASE-1 VERSION OF PROGRAM

1. Program Output

a. Data Plots

The data program is designed to produce multiple plots of the data in conjunction with a cathode-ray-tube plotter. The program does not operate on line with the plotter but produces a tape that is fed into the plotter as a separate operation. The data-tape format is compatible with either the Stromberg-Carlson 4020 plotter or the Benson-Lerner BL-120 unit. NRO has obtained a Benson-Lerner unit that will be used to plot the X-engine test data.

Plots can be made for any individual channel, various combinations of channels, or for any parameters computed by the program. The basic plot will be 10 by 10 in., with a 10-line/in. grid. Time will be plotted on the horizontal axis and can be plotted to any desired range over the 10-in. space. The vertical axis will provide for three separate, independent scales, which can be different ranges for the same parameter or entirely different parameters. Sample plots are shown on Figures 1 and 2.

The channels plotted on each scale can be arranged in groups of up to ten channels with a title. The advantage of the group arrangement is that channels that measure a common parameter can be grouped together under one title. The name for each group is a program input and should be designed to define the measurement for analysts using the plots.

Up to ten groups can be shown on each of the scales. However, there can be no more than 32 curves shown on any one plot, and the number of curves per group is reduced if a larger number of groups is to be shown. Generally, a large number of curves is shown on a plot only if the curves are measuring a common parameter, and the reduced number of curves should not be a practical limitation.

Figure 1 - Sample Plot, Duct Water Pressure at Inlet and Outlet

Figure 2 - Sample Plot, Operating Conditions in Primary Chamber and at Primary Nozzle Exit

If a scale is specified that is out of the range of the actual data, the program will not plot anything. Similarly, if data are missing in the middle of a curve, the program will plot a discontinuous line.

Each plot has a code number that relates it to a subject of analysis. For instance, all plots dealing with the turbine could be assigned a T code number that would be initiated with T-1 and continue until all plots under this subject were exhausted. This code number is the key to organizing the plots in the data book produced from the program. The advantage of this system is that all plots relating to a given subject are grouped together.

If a plot or a channel has various subjects under which it will be analyzed, it can be provided as input so it appears under all subjects where it is needed for analysis. The code number appears in the title of the plot and in the lower right-hand corner. The lower right-hand corner code is provided so a plot can be easily located when a person is going through the data book.

Two indexes are provided along with the plots. The first is a subject index that is arranged alphabetically according to the code number. If an analyst were interested in turbine inlet temperatures, he would look under the subject of turbine until he located a plot with temperatures. Each plot has a title that indicates the material on the plot. The code number would be shown for this plot, and the plot could be located from this number. There is also an alphabetic channel index. If an analyst is interested in a specific channel, the plot or plots on which this channel is shown will be indicated.

A maximum of four time intervals can be plotted for any run. For each time interval, a maximum of 500 plots can be obtained with a maximum of 32 curves per plot. Each plot interval can have completely different plots. For instance, during a typical run, it would be desirable to

plot all channels for the entire run period on one 10-in. plot to show capsule summaries of the run. To complement this plot, it may be desirable to show a greatly expanded blowup of bootstrap for limited channels, a medium blowup of the startup ramp, and a similarly scaled plot for shutdown. The steady-state period could then be handled in detail by the steady-state summaries.

Each plot will always be made up from 500 points that are joined into a continuous line. The program automatically takes the time period to be plotted and determines the sampling rate required to get 500 points in the period. This operation is independent of the main-program data processing, which has a sampling rate determined by its input. For a 15-min run, if the program were processing data at 0.1-sec intervals, it would process $15 \times 600 = 9000$ time periods to plot the entire run. The plot routine would sample every eighteenth point ($18 \times 500 = 9000$) to prepare an overall run summary plot.

Five-hundred points per plot were selected because they will appear as a solid line on 10 in. of grid and any higher rate of sampling would be meaningless. The program user must recognize, however, that the time scale determines the frequency of events that can be detected from any plot.

Data selected by the plot program are stored on disk in a plot-oriented arrangement; i.e., time-1 data for plot 1 through plot X, then time-2 data for plot 1 through plot X, and continuing in that pattern. If the maximum number of curves were being plotted, there would be 32,000,000 words of data stored on the disks. The data are sorted into continuous channel format in a series of 380,000 IBM 360-65 bites into core. Preliminary runs indicate that this sort can be completed in about 15 min of computer time.

After sorting, approximately 2 sec per curve is required to produce a tape for driving the CRT plotter. If each channel were plotted one time, the preparation of the plot tape would take approximately 0.6 hr. A similar length of time would be required for microfilm preparation on the CRT plotter.

b. Steady-State Listings

Steady-state listings may be obtained for up to five time periods on any run. These summaries may be sequential, gapped, overlapping, or some fully contained within others. In other words, if there is an extended hold period, the entire period can be averaged with up to four smaller periods being averaged within the long period. The smaller periods can also be within each other.

The output is displayed alphanumerically within groups called "units of analysis." The units of analysis for X-engine testing are listed in Table 1. The object of this type of listing is to group in one location the channels that are of interest to a particular engineer. The arrangement of output is controlled by input cards and must be determined as a part of the pretest planning.

The accumulated steady-state data are stored on disks after the main program has performed the necessary calculations. All channels are automatically placed in disk storage for each time period averaged by the program. However, all channels need not be printed out. An option permits the printout of the channels within any unit of analysis desired (see Table 1).

The disk used to accumulate the data is set aside for the NRO data program, and this information is not erased until the data program is operated again. If a permanent record is desired, the steady-state information can be written on tape.

c. Data Anomalies

The program has two features to flag discrepant data. On plots, if data are missing or the NTO engineering-unit tape has a negative zero in a data slot (NTO format for indicating bad data), the program will skip the missing data on the plot.

TABLE 1

UNITS OF ANALYSIS FOR X-ENGINE TESTING

Steady-state channels are to be printed out in accordance with the following units of analysis.

1	LH ₂ Feed System
2	Pump and Pump-Discharge System
3	Nozzle
4	Reflector
5	Shield and Dome Shield
6	Thrust Chamber
7	Turbine Feed System
8	Turbine and Turbine Exhaust
9	Duct
10	Steam Generators
11	External Shield and Test Cell
12	LH ₂ Cooldown System
13	GH ₂ System
14	GHe System
15	Control Systems
16	Miscellaneous
17	Core

In addition, a listing is prepared that includes all data channels processed by the program. The list summarizes the total number of data points on each channel that were processed by the program, as well as how many were missing or had the negative zero notation. This information does not give the time at which the discrepancy occurred, but it does flag channels that have multiple discrepancies and which should be investigated further.

The program also prints out a listing of any flows that could not be computed in the flow subroutine. This information is printed out on line in the event the monitoring engineer desires to take corrective action.

d. Events Listing

The program will automatically print out a listing for all events on the data tapes if this option is used.

The printout will contain the following items:

1. The title (e.g., X-Engine Test Events Data).
2. The control-room start and end times (CRT times) checked for event changes.
3. The EP number.
4. The data of the test.
5. A list of the events; i.e., the channel numbers, including
 - a. The test-event description (i.e., the channel title) followed by an (S) or (V) to denote a switch or valve, respectively.
 - b. The status ("on" or "off" for a switch, or "open" or "closed" for a valve) of each event at the start of data processing.
 - c. The control-room time (CRT) for each event change.

The number of changes in an event that will be listed is limited to 11. This limitation is imposed to prevent an intermittent channel from producing an excess volume of paper. Most events are limited in the number of times they should occur during a run, and 11 was arbitrarily selected as a large number.

The order in which the events are listed is determined by their order in the NTO EU deck. Hence, an alphameric listing should result. All items are printed out as a result of the initial run.

The total events listing is only obtained the first time the events are checked. Only changes are noted during subsequent checks. If more than one run is being analyzed during the day in one continuous data-processing run, care should be taken to recognize that, the first time the program reads data from the second run, it will assume that all noted events changes occurred at that time. If the events were changed in an intervening period, where no data were being recorded, the program would not recognize this.

2. Data Input

a. Tape-Read Routines

The data analysis program is designed to run with tapes obtained from the ETS-1 data system. The tape-read routines are compatible with the N-3 engineering-unit tapes produced on the NTO CDC-3200 computer. The format of these tapes is described in Appendix C of this report.

The engineering-unit tapes are corrected for calibration errors by NTO, and no provisions are made for automatic correction of any data read into the program. The various mathematical routines described later in this report will allow a channel to be corrected, but this correction can only be made with input cards. The ETS-1 data system has two tape outputs that

simultaneously produce data from two subsystems. The data program is designed to operate with a tape from one system or to read both subsystems simultaneously. The data program produces one tape that includes the data from the two NTO systems as well as any newly computed parameters. This tape can be read back into the program and handled as if it were new data.

When operating on any data tapes, the program must also have a card deck that relates a channel to a word number on the tape. All input is handled by channel number, and this information is not on the tapes received from NTO.

The program can read data at 10 samples/sec or slower. The 100- and 50-sample/sec data on the tapes are reduced to 10-sample/sec frequency when the program reads only the first word encountered on a data frame. Up to three discontinuous time periods, equivalent to three separate tests, can be processed during a program run; or the sampling rate can be changed up to five times during a run. There is no limit on the length of run that can be processed by the program. There are limits on the various output features, however, which could limit the program's ability to display the results from a run that was over 1-hr duration.

In addition to processing the 10-sample/sec data, the program can read the events data that are recorded on the NTO tapes. To read this information, the program requires the inputting of a card deck that relates the events to the tape location. No prepared input is needed to tabulate the events information.

The tapes are in CDC-3200 floating-point format, which is not compatible with the IBM 360-65 floating-point format. A description of the routine that converts the data is included as Appendix D.

b. Check of Channels Used in Pretest Planning

This routine compares the channels on the NTO card deck with the channels on the NRO master deck. A listing is obtained that prints out channels that are present on only one of the decks. The routine is operated independently of the main program to check the NRO planning against the NTO channels actually wired for a given test.

Normally, NTO makes a data-system check about three days before a run. At this time, a card deck can be prepared to check the NRO master deck that reflects the planning for the given run. The discrepancies can be reviewed to determine if some of the plots or calculations require revision because of the actual channels to be used. This same procedure can be repeated after run day to compare the as-run NTO deck with the NRO master deck.

3. Data Calculations

a. Raw-Data Smoothing

A data-smoothing routine was included in the program so that noise could be eliminated that is frequently present on channels. Any channel can be smoothed with this routine, but a channel cannot be retained in both the smoothed and unsmoothed conditions.

A seven-point data-smoothing routine was selected for the X-engine program on the basis of the characteristics of the data-acquisition system and the normal frequencies analyzed with the narrow-band data.

The smoothing routine considers three points on either side of the point to be smoothed. The data are smoothed by using a formula that is based on a least-squares third-degree polynomial curve fit.

$$Y = \frac{-2Y(-3) + 3Y(-2) + 6Y(-1) + 7Y(0) + 6Y(1) + 3Y(2) - 2Y(3)}{21} \quad (1)$$

The mean and standard deviation are calculated for the seven points. Should any point lie outside the three-standard-deviation limit, it is rejected. The smoothed value (Y) is then computed with the rejected point removed from Equation 1 and its weighting removed from the denominator. Should more than three of the seven points be rejected, the routine will revert to the observed data point.

In a check of the smoothing routine, two types of data were used as input. The first type included sine- and square-wave functions and designed cases. The second type included actual data from the CFDTS and NRX/EST tests.

A sine wave was programmed and values were selected at 0.1-sec intervals. These values were smoothed by using Equation 1. Frequencies of the input sine wave and the smoothed wave were consistent up to 5 cps. Above 5 cps, the data were aliased by the sampling rate to a frequency of less than 5 cps, and the smoothing routine did not affect the aliased frequency. The amplitudes of the input sine wave and the output smoothed wave were consistent through 0.5 cps; thereafter, the output amplitude decreased as the frequency increased until 5 cps, where it was completely suppressed (see Figure 3).

Raw test data and smoothed data from CFDTS and NRX/EST runs are shown in Figures 4 through 6. Figure 4 shows a plot of smoothed data versus raw data for data from CFDTS Run No. 19. Figures 5 and 6 show how the smoothing routine follows the oscillations that were observed at low power during NRX/EST tests. Turbopump speed oscillation of the type occurring during dry pump starts are low frequency (less than 0.5 cps) and consequently will not be filtered by the smoothing routine.

NOTE:

ABOVE 5 CPS,
FREQUENCIES ARE
ALIASED AND
AMPLITUDE IS
VARIED

Figure 3 - Smoothing Routine, Characteristic Sine-Wave Input

Figure 4 - CFDTS Data, Run 19

Figure 5 - NRX/EST EP IIC Data, Turbopump Speed

Figure 6 - NRX/EST EP IIC Data, Reflector Inlet Plenum Pressure

The ETS-1 data system has a 4-cps filter in the high-level system that eliminates high frequencies, and the low-level 10-sample/sec system aliases any frequencies above 5 cps. Thus, the smoothing routine in essence dampens frequencies from 1 to 5 cps. It also dampens the aliased results observed from the higher frequencies. It rejects spurious points that are not related to the bulk of the data.

b. Channel Averaging

A channel-averaging routine is available that permits the averaging of any number of channels. The channels are linearly weight-averaged, and the average value obtained can be used for plotting or for any subsequent routines in the program. The averaged channel is assigned an eight-character alphanumeric code number and a title which are similar to all data channels. The averaging routine was separated from the other mathematical routines to permit an unlimited number of channels to be averaged.

c. Flow Calculations

A special routine was provided to automatically compute the parameters that are common to all engine tests.

The flow-calculation subroutine consists of 36 flow calculations and one thermal-power calculation equation. All of these equations are coded into the program, and changes of constants and channel numbers must be handled with override cards. To plot the various flows that are automatically computed, a plot card must be prepared as with any other raw-data channel. If any flow cannot be computed, this information is printed on line to the monitoring engineer.

Many of the flows in the engine system are computed by more than one method. To be useful to the data analyst, the formulas used for each of the flow calculations must be available. These formulas are itemized in Figure 7.

FLOW CALCULATIONS

1. PUMP INLET FLOW RATE

$$W \text{ PUMPIN} = (\text{FE 014}) (0.4970) \text{ lb/sec}$$

$$\text{Where density, } (\rho) = f[(\text{TT103}), (\text{PT104})]$$

$$W \text{ PMPIN1} = W \text{ NOZ} + W \text{ TURB } \phi, \text{ lb/sec}$$

2. NOZZLE FLOW RATE

$$W \text{ NOZ} = \frac{AV \text{ PCHAM}}{\sqrt{AV \text{ TCHAM}}} [(A)(0.1433 - 1.76 \times 10^{-6})(AV \text{ TCHAM})], \text{ lb/sec}$$

$$W \text{ NOZ1} = W \text{ PUMPIN} - W \text{ TURB } \phi, \text{ lb/sec}$$

3. TURBINE INLET FLOW RATE

$$A. \quad W \text{ TURB 1} = f(Y) K_3 \sqrt{\rho(\text{PD804})}, \text{ lb/sec}$$

$$\text{Where density, } (\rho) = f[(AV \text{ TTURI}), (\text{PT814})] \text{ H}_2 \text{ Properties}$$

$$K_3 = 3.031, \text{ Flow Constant}$$

$$f(Y) = 1.0 - (0.5383 x + 0.2167 x^2)$$

$$\text{Where } X = \frac{\text{PD804}}{\text{PT814}}$$

$$B. \quad W \text{ TURBI1} = \frac{K (AV \text{ PTURI})}{\sqrt{(AV \text{ TTURI})}}, \text{ lb/sec}$$

$$K = \text{Choke Flow Constant}$$

$$= 0.525 + 7.6 (AV \text{ TTURI}) (10^{-5})$$

$$C. \quad W \text{ TURBI2} = (W \text{ C DIL}) \frac{(h_{\text{NOZ}} - h_{\text{DIL}})}{(h_{\text{NOZ}} - h_{\text{TURB}})}, \text{ lb/sec}$$

$$\text{Where enthalpy} = h, (\text{Btu/lb}) = f(T, P), \text{ H}_2 \text{ Property}$$

$$h_{\text{NOZ}} = [T = (AV \text{ TCHAM}), P = (AV \text{ PCHAM})]$$

$$h_{\text{DIL}} = [T = (\text{TT227}), P = (\text{PT226})]$$

$$h_{\text{TURB}} = [T = (AV \text{ TTURI}), P = (AV \text{ PTURI})]$$

$$D. \quad W \text{ TURBI3} = (W \text{ TURB } \phi) - (W \text{ BE CL}), \text{ lb/sec}$$

Figure 7 - Flow Calculations
(Sheet 1 of 4)

$$E. \quad W \text{ TURBI4} = (W \text{ PUMP } \emptyset) - (W \text{ NOZ}), \text{ lb/sec}$$

4. DILUENT FLOW RATE

$$A. \quad W \text{ C DIL} = f(Y) K_2 \sqrt{\rho(PD209)}, \text{ lb/sec}$$

$$K_2 = 0.57, \text{ Flow Constant}$$

$$\text{Density} = \rho, (\text{lb/ft}^3) = f[(TT227), (PT226)], H_2 \text{ Property}$$

$$X = \frac{PD209}{PT226} \text{ in } f(Y) \text{ Equation, Section 3.a}$$

$$B. \quad W \text{ C DILI} = (W \text{ TURB I}) \frac{(h_{\text{NOZ}} - h_{\text{TURB}})}{(h_{\text{NOZ}} - h_{\text{DIL}})}, \text{ lb/sec}$$

$$\text{Enthalpy} = h, \text{ Btu/lb} - \text{ see Section 3.c for pressure and temperature.}$$

5. TURBINE OUTLET FLOW RATE

$$A. \quad W \text{ TURB } \emptyset = K_5 \frac{(PT883)}{\sqrt{TE846}} + \frac{K_6 (PT885)}{\sqrt{TE847}}, \text{ lb/sec}$$

$$K_5, K_6 = \text{Flow Constant} = 3.358$$

$$B. \quad W \text{ TURB } \emptyset 1 = (W \text{ TURB I}) + (W \text{ BE CL}), \text{ lb/sec}$$

6. BEARING COOLANT FLOW RATE

$$A. \quad W \text{ BE CL} = 1.41 \frac{(AV \text{ PPMP } \emptyset)}{1000.0} - 0.78 \left[\frac{AV \text{ PPMP } \emptyset}{1000.0} \right]^2, \text{ lb/sec}$$

$$B. \quad W \text{ BECL1} = (W \text{ TURB } \emptyset) - (W \text{ TURB I}), \text{ lb/sec}$$

$$C. \quad W \text{ BECL2} = (W \text{ PUMPIN}) - [(W \text{ NOZ}) + (W \text{ TURB I})], \text{ lb/sec}$$

Figure 7 - Flow Calculations
(Sheet 2 of 4)

7. BOLT COOLANT FLOW RATE

$$W_{BLT CL} = f(Y) K_1 \sqrt{\rho(PD163)}$$

Where $K_1 = 0.452$, Flow Constant

Density = $\rho = f[(AV_{TNTIN}), (PT172)], H_2$ Property

$$X = \frac{PD163}{PT172} \text{ in } f(Y) \text{ Equation 3(a)}$$

8. PUMP DISCHARGE FLOW RATE

$$W_{PUMP \emptyset} = (W_{PUMPIN}) - (W_{BE CL}), \text{ lb/sec}$$

9. HOT BLEED PORT FLOW RATE

$$A. \quad W_{HBP} = (W_{TURB I}) - (W_{C DIL}), \text{ lb/sec}$$

$$B. \quad W_{HBP1} = \frac{(W_{TURB I}) h_{TURB} - (W_{C DIL}) (h_{DIL})}{h_{NOZ}} \text{ lb/sec}$$

Where h = enthalpy, see Equation 3(C)

10. ENGINE THERMAL POWER

$$POWER = [(W_{NOZ}) (h_{NOZ} - h_{RFL}) + (W_{TURB I}) (h_{TURB} - h_{RFL})] (1.055)(10)^{-3},$$

Megawatts

Where h = enthalpy, Btu/lb

h_{NOZ}, h_{TURB} , See Equation 3(C)

$h_{RFL} = f[(AV_{TRFIN}), (AV_{PRFIN})^2], H_2$ Property

11. DUCT FLOWS

<u>Channel Number</u>	<u>Description</u>	<u>Equation</u>
W DUCI 1	Duct Section 1 left water inlet flow =	$1060\sqrt{FT054..F}$
W DUCI1A	Duct Section 1 right water inlet flow =	$1060\sqrt{FT055..F}$
W DUCI 2	Duct Section 2 left water inlet flow =	$1405\sqrt{FT056..F}$

Figure 7 - Flow Calculations
(Sheet 3 of 4)

<u>Channel Number</u>	<u>Description</u>	<u>Equation</u>
W DUCI2A	Duct Section 2 right water inlet flow =	1405 $\sqrt{\text{FT057} \dots \text{F}}$
W ELJI 1	Duct Section 2 jacket water inlet flow =	198.5 $\sqrt{\text{FT060} \dots \text{F}}$
W DUCI 3	Duct Section 3 left water inlet flow =	1290 $\sqrt{\text{FT058} \dots \text{F}}$
W DUCI3A	Duct Section 3 right water inlet flow =	1290 $\sqrt{\text{FT059} \dots \text{F}}$
W SG W 1	Steam-generator injection-water flow =	3.33 $\sqrt{\text{FT406W} \dots \text{F}}$
W SG W 2	Steam-generator injection-water flow =	3.33 $\sqrt{\text{FT406X} \dots \text{F}}$
W SG W 3	Steam-generator injection-water flow =	3.33 $\sqrt{\text{FT406Y} \dots \text{F}}$
W SG F 1	Steam-generator fuel flow = (0.0166)(FT405W.F) - (0.03125)(TT478..F) + 17.961	
W SG F 2	Steam-generator fuel flow = (0.0166)(FT405X.F) - (0.03125)(TT478..F) + 17.961	
W SG F 3	Steam-generator fuel flow = (0.0166)(FT405Y.F) - (0.03125)(TT478..F) + 17.961	
W SG Ø 1	Steam-generator liquid-oxygen flow = (0.0276)(FT409W.F) - (0.03666)(TT479..F) + 17.432	
W SG Ø 2	Steam-generator liquid-oxygen flow = (0.0276)(FT409X.F) - (0.03666)(TT479..F) + 17.432	
W SG Ø 3	Steam-generator liquid-oxygen flow = (0.0276)(FT409Y.F) - (0.03666)(TT479..F) + 17.432	

Figure 7 - Flow Calculations
(Sheet 4 of 4)

The power routine currently in the program is not exact during transients, as it does not include corrections for the engine sensible heat.

d. Special Calculations

The previous sections described special mathematical routines that are used regularly in data analysis. To increase the versatility of the program, a special mathematical computation technique was incorporated to permit general-type calculations. This technique makes it feasible to perform a wide variety of mathematical operations on the test data without recompiling the program. These operations are controlled by simple card input.

Two general mathematical equations are coded into the program, and, at the time of compilation, they have no reference to specific channels of data or constants. The equations are:

$$A. \quad ZNEW = \frac{\Delta_1 (\Delta_2 + \Delta_3)^{\Delta_6}}{(\Delta_4 + \Delta_5)^{\Delta_7}}$$

and

$$B. \quad ZNEW = \Delta_2 + \Delta_3 (\Delta_1)^1 + \Delta_4 (\Delta_1)^2 + \Delta_5 (\Delta_1)^3 + \Delta_6 (\Delta_1)^4 + \Delta_7 (\Delta_1)^5.$$

In either of the equations, the delta (Δ) values may be fixed (constants - including zero) or may be channel data. Any combination of each is acceptable for a specific application. When this option is used, the newly created data are handled as any other channel data and can be plotted or printed out as required.

Examination of the two mathematical expressions shows that a wide variety of calculations can be performed simply and without the need to recompile the entire program. Some simple examples that may be considered are:

1. Addition of a constant to channel data, as might be necessary to correct psig to psia or other units.

EXAMPLE: $ABSPRESS = 12.72 + PT128..A$,
where the input to Equation A is:

$$\begin{aligned}\Delta_1 &= 1.0, \Delta_2 = 12.72, \Delta_3 = PT128..A, \\ \Delta_4 &= 0, \Delta_5 = 1.0, \Delta_6 = 1.0, \text{ and } \Delta_7 = 1.0.\end{aligned}$$

2. Multiplication of a constant by a channel data that has been raised to a power that is necessary to calculate flow by means of venturi calculations.

EXAMPLE: $CALCFLOW = 1.316 \sqrt{PT172..A}$,
where the input to Equation A is:

$$\begin{aligned}\Delta_1 &= 1.316, \Delta_2 = 0, \Delta_3 = PT172..A, \\ \Delta_4 &= 0, \Delta_5 = 1.0, \Delta_6 = 0.5, \text{ and } \Delta_7 = 1.0.\end{aligned}$$

3. Multiplication of one channel value by another raised to a power and then divided by another channel value as necessary to determine flow coefficients.

EXAMPLE: $FLOWCORE = (WTURBI) \frac{TE846..A}{PT883..A}$
where the input to Equation A is:

$$\begin{aligned}\Delta_1 &= WTURBI, \Delta_2 = 0, \Delta_3 = TE846..A, \\ \Delta_4 &= 0, \Delta_5 = PT883..A, \Delta_6 = 0.5, \text{ and } \\ \Delta_7 &= 1.0.\end{aligned}$$

4. The polynomial allows a correction to the data to be made in accordance with the standard method.

EXAMPLE: $POLYF..F = -3.7214 + (0.1213)(FT406X.F) + (0.9217)(FT406X.F)^2 + (1.0032)(FT406X.F)^3 + (0.0432)(FT406X.F)^4 - (0.3217)(FT406X.F)^5$,
where the input to Equation B is:

$$\begin{aligned}\Delta_1 &= FT406X.F, \Delta_2 = -3.7214, \Delta_3 = 0.1213, \\ \Delta_4 &= 0.9217, \Delta_5 = 1.0032, \Delta_6 = 0.0432, \text{ and} \\ \Delta_7 &= -0.3217.\end{aligned}$$

5. Curve matches can be made and input via the polynomial such as would be necessary to calculate stress from strain via a specific stress-strain curve match.

EXAMPLE: $CALSTRES = 26.1520 + (3.052)(MG102L.A) + (0.2615)(MG102L.A)^3$,
where the input to Equation B is:

$$\begin{aligned}\Delta_1 &= MG102L.A., \Delta_2 = 26.1520, \Delta_3 = 3.052, \\ \Delta_4 &= 0.0, \Delta_5 = 0.2615, \Delta_6 = 0.0, \text{ and } \Delta_7 = 0.0.\end{aligned}$$

4. Retention of Data

The program produces a time-based data tape in IBM 360-65 language that includes all data taken from the NTO tapes and any new parameters that have been computed by the program. These tapes will be stored and can be used for additional plotting on data calculations as desired. A plot tape is produced that is used to drive the CRT plotter. This tape will be destroyed as all plots are automatically recorded on microfilm by the CRT plotter.

The steady-state and plot data placed on the disks are retained by the IBM 360-65 system. The disks have been set aside for the data program and these cannot be used by other programs. The disk data will be retained until the program is operated again. At this time the disks will be erased.

BLANK

APPENDIX A

CARD INPUT FORMAT

Input-Card Instructions

A diagram showing the major blocks of card input to operate the data program is shown in Figure 1. The program deck illustrated at the bottom of Figure 1 contains the control cards for operating the Aerojet IBM 360-65 system. These cards are maintained in Aerojet's Computing Sciences Library and are obtained by calling for Program No. D-57101, the NRO data program. The master deck (see Figure 1) contains a description of all channels to be processed, which can be input either as cards or on tape. The NTO engineering-unit deck shown next must be input as cards. This deck, which is obtained from NTO along with the data tapes, relates the channels to the word numbers on the tape.

The main program processing cards (see Figure 1) include all instructions for plotting, averaging, smoothing, special calculations, and steady-state computations. These instructions can be input as cards or on tape. The override cards correct cards in the previous blocks. These cards must be input as cards but can be put into their proper place by preparing a new input tape after they have been processed into the computer. The plot program instructions are used to prepare the plot tape from the time-oriented data processed by the main program. These instructions can be input as cards only. The steady-state printout control cards cause the steady-state data on disks to be printed out in alphameric format.

The program can be operated four ways. When cross-checking the NRO master deck with the NTO deck, the program will only read cards through the NTO deck. It can either be input with these cards or with the entire input tape, which will include additional information that will not be used.

The program can also be operated through the main program, which will perform all calculations, store plot data and steady-state data on disks, and

Figure 1 - Major Blocks of Card Input

Appendix A

create a new data tape. To use this option, all previous decks in Figure 1 must be input, and the plot program instructions may or may not be included.

The plot program may be used in conjunction with the main program in one continuous run or it may be run separately. To run it separately, data must be on disks in the proper format, and the program deck must be included in front of the plot deck. Similar to the plot program, the steady-state printout may be obtained in one continuous run with the main program or may be input separately after the data have been stored on disks.

The following sections describe in detail the card format for all decks except the program deck, which is maintained by Computing Sciences, and the NTO deck, which is prepared at NTO.

NRO Master Deck (Block No. 2, Figure 1)

All cards in the master deck have the same format. There should be one card for each channel in the test being processed. The master deck may be input as a tape or on cards. The card format is shown in Table 1.

NTO Engineering-Unit Deck

The format for these cards is controlled by NTO. A description is presented in Appendix B.

Main Program Processing Deck (Block No. 4, Figure 1)

A schematic for the cards in this deck is shown in Figure 2. The section designated "control cards" gives instructions for processing individual channels or for special calculations to be performed. Each subheading in this section is sequential; i.e., all AVGT cards are completed before FLOW cards are started.

Appendix A

TABLE 1

NRO Master Deck

<u>Card Column</u>	<u>Entry and Explanation</u>
1 - 8	Channel number.
9 - 10	Blank.
11 - 41	Title of channel. This title will be used on plots and summaries where this channel is to be displayed. Generally, the MRL title should be used for consistency.
42 - 50	Units of parameter (psi, °R, etc.).
51 - 65	Blank.
66 - 67	These columns are used to identify under which summary heading the channel should be displayed in steady-state printouts (see Section III.B.1.b of main-text writeup).
68 - 70	Sequence in the steady-state printout. The program will not make a sort, and the desired printout order must be indicated.
71	Blank.
72	1 if channel is to be smoothed. 0 if channel is not to be smoothed.
73 - 80	Blank.

Appendix A

Figure 2 - Card Setup for Main-Program Processing Deck

Appendix A

The plot cards are used to write the data on disks in the proper format for plotting. The card order is controlled by the requirements for specifying the plots; and the groups of cards shown will not be physically located together. For instance, a P-3 card specifies a time interval to be plotted. The P-4 through P-10 cards contain instructions for all plots that will be made during this time interval. All P-4 through P-10 cards necessary to define the plots in time interval No. 1 will be physically located behind the first P-3 card before the second P-3 card is placed in the deck. This order must be maintained or the program will not properly prepare the data for plotting.

The following paragraphs describe in detail the information that must be put on each card.

A Test Series Card (Card No. 1, Figure 2)

This card defines the test series for which the tapes are to be read. This information will be used on the various plots and summaries. The card format is shown in Table 2.

Tape Card (Card No. 2, Figure 2)

This card defines the time intervals that will be read from the tapes for data processing. The program will read data for up to five different time intervals; i.e., if it is not desired to process all data or if there were several runs during a day, these can all be handled with the input instructions on this card. The program can read data from one subsystem, two subsystems, or a tape created by the data program. This information is also coded on the tape card (see Table 3).

Appendix A

TABLE 2

TEST-SERIES CARD

<u>Columns</u>	<u>Format</u>	<u>Description</u>
1 - 16	Alphameric	Used to identify the test series.
17	Blank	
18 - 33	Alphameric	Used to denote test-firing date (e.g., 19 March 1967).
34 - 38	Alphameric	Time of test.
39 - 42	Integer	Number of NTO cards included in the input deck.
43 - 45	Integer	Number of NTO tapes to be read, one system only.
46 - 48	Integer	If nonzero, no plot requirement.
49 - 51	Integer	If nonzero, test-events data will be compiled; otherwise these data will be ignored.
52 - 80	Blank	-

Appendix A

TABLE 3

CARD INPUT FORMAT

TAPE Card (Card No. 2, Figure 2)

<u>Card Columns</u>	<u>Entry</u>
1 - 4	TAPE.
5 - 6	Blank.
7 - 8	11 if reference is to one NTO EU tape; 0 or blank, if reference is to two NTO EU tapes. 1 if reference is to IBM 360-65 OS test data tape created on previous run.
9 - 14	An integer* in tenths of a second representing the first control-room-time (CRT) start time.
15	Blank.
16 - 19	An integer in tenths of a second representing the sampling rate between the first and second CRT start times.
20 - 25	An integer in tenths of a second representing the second CRT start time.
26	Blank.
27 - 30	An integer in tenths of a second representing the sampling rate between the second and third start times.
31 - 36	An integer in tenths of a second representing the third CRT start time.
37	Blank.
38 - 41	An integer in tenths of a second representing the sampling rate between the third and fourth start times.
42 - 47	An integer in tenths of a second representing the fourth CRT start time.

* All integers are right-adjusted.

Appendix A

TABLE 3 (cont.)

<u>Card Columns</u>	<u>Entry</u>
48	Blank.
49 - 52	An integer in tenths of a second representing the sampling rate between the fourth and fifth start times.
53 - 58	An integer in tenths of a second representing the fifth CRT start time.
59	Blank.
60 - 63	An integer in tenths of a second representing sampling rate from beginning to end of the fifth start time.
64-80	Blank.

Appendix A

SMTH Cards (Card No. 3, Figure 1)

These cards control the instructions for smoothing channels. A card must be prepared for each channel that is to be smoothed. A channel can be alternately smoothed or processed as raw data, if desired. This can be accomplished a maximum of five times on any channel during one data run. The smoothing periods need not be the same for all channels, as each channel has individual instructions. Details of input are shown in Table 4.

AVGT Cards (Card No. 4, Figure 1)

These cards control the instructions for the steady-state periods that are to be averaged during the run. All channels in the master deck will be averaged over the time periods defined on this card. A maximum of five time periods can be summarized during any one run. The time periods are completely independent and can be overlapping or one can be completely within another, if desired. An average can not be obtained over discontinuous data periods. If there has been a shutdown with a data discontinuity, it is impossible to average a period including both runs. Details for input are shown in Table 5.

Flow Cards (Card No. 5, Figure 2)

The flow calculations are unique in the data program in that they are coded into the program in a fixed format. During each run, the program will attempt to compute the flows from the built-in equations. The channels to be used are also built in. If the flows are not to be computed or if it is desired to change a constant or a channel to be used, a flow override card must be input. These cards will change the subject flow calculation. Details of the flow routine are described in Section III.B.3.c of the main text writeup, and the format for the override cards is shown on Table 6.

Appendix A

TABLE 4

SMTH CARD

<u>Columns</u>	<u>Format</u>	<u>Description</u>
1 - 4	SMTH	Control card for smoothing routine.
5 - 6	Blank	
7 - 12	Integer	First* CRT start time in tenths of a second for smoothing.
13 - 18	Integer	First CRT stop time in tenths of a second for smoothing.
19 - 24	Integer	Second CRT start time in tenths of a second for smoothing.
25 - 30	Integer	Second CRT stop time in tenths of a second for smoothing.
31 - 36	Integer	Third CRT start time in tenths of a second for smoothing.
37 - 42	Integer	Third CRT stop time in tenths of a second for smoothing.
43 - 48	Integer	Fourth CRT start time in tenths of a second for smoothing.
49 - 54	Integer	Fourth CRT stop time in tenths of a second for smoothing.
55 - 60	Integer	Fifth CRT start time in tenths of a second for smoothing.
61 - 66	Integer	Fifth CRT stop time in tenths of a second for smoothing.
67 - 80	Blank	

* All time periods are nonoverlapping and are in ascending order.

Appendix A

TABLE 5

AVGT CARD

<u>Columns</u>	<u>Format</u>	<u>Description</u>
1 - 4	AVGT	(Control card for steady-state hold)
5 - 6	Blank	
7 - 12	Integer	First CRT start time in tenths of a second for averaging
13 - 18	Integer	First CRT stop time in tenths of a second for averaging.
19 - 24	Integer	Second CRT start time in tenths of a second for averaging.
25 - 30	Integer	Second CRT stop time in tenths of a second for averaging.
31 - 36	Integer	Third CRT start time in tenths of a second for averaging.
37 - 42	Integer	Third CRT stop time in tenths of a second for averaging.
43 - 48	Integer	Fourth CRT start time in tenths of a second for averaging.
49 - 54	Integer	Fourth CRT stop time in tenths of a second for averaging.
55 - 60	Integer	Fifth CRT start time in tenths of a second for averaging.
61 - 67	Integer	Fifth CRT stop time in tenths of a second for averaging.
67 - 80	Integer	Blank.

Appendix A

TABLE 6

FLOW CARD

<u>Columns</u>	<u>Entry</u>
1 - 4	FLOW.
5	Blank.
	Reference is made to specific flow equations.
6 - 7	01 - if $\dot{W}_{BC}$ is required.
6 - 7	02 - if $\dot{W}_{TIN}$ is required.
6 - 7	03 - if $\dot{W}_{CDIL}$ is required.
6 - 7	04 - if $\dot{W}_N$ is required.
6 - 7	05 - if $\dot{W}_{TOUT1}$ is required.
6 - 7	06 - if $\dot{W}_{TOUT2}$ is required.
6 - 7	07 - if $\dot{W}_{TIN1}$ is required.
6 - 7	08 - for Enthalpy (to calculate core power).
6 - 7	09 - if $\dot{W}_{BECL}$ is required.
6 - 7	10 - if W SG F 1 is required.
6 - 7	11 - if W SG F 2 is required.
6 - 7	12 - if W SG F 3 is required.
6 - 7	13 - if W SG Ø 1 is required.
6 - 7	14 - if W SG Ø 2 is required.
6 - 7	15 - if W SG Ø 3 is required.
6 - 7	21 - if W DUCI 1 is required.
6 - 7	22 - if W DUCI 1A is required.
6 - 7	23 - if W DUCI 2 is required.
6 - 7	24 - if W DUCI 2A is required.

Appendix A

TABLE 6 (cont.)

<u>Columns</u>	<u>Entry</u>
6 - 7	25 - if W DUCI 3 is required.
6 - 7	26 - if W DUCI 3A is required.
6 - 7	27 - if W ELJI is required.
6 - 7	28 - if W SG W 1 is required.
6 - 7	29 - if W SG W 2 is required.
6 - 7	30 - if W SG W 3 is required.
8	Blank.
9 - 18	Decimal integer - flow constant, FL(K); based on venturi design for given test.
19 - 38	Blank.
39 - 48	Blank, if no override* required; channel number, left-adjusted, for pressure.
49 - 56	Blank, if no override required; channel number, left-adjusted, for temperature.
57 - 64	Blank, if no override required; channel number, left-adjusted, for DELTA PRESSURE.
65 - 70	Blank.
71 - 72	99 if flow equations are to be ignored; otherwise blank.
73 - 80	Blank.

* In order to accommodate channel-number changes up to actual test run time, the override feature is used to override (that is, replace) the channel number built into the program by the current channel measuring the specified pressure, differential pressure, or temperature.

Appendix A

Z New Card (Cards No. 6 and 7, Figure 2)

These cards are used to control the routine that performs the generalized calculations. There are two basic equations in this routine, namely a fifth order polynomial and a basic addition, subtraction, multiplication, division, and power equation. The equations are controlled by a series of numbers that are defined on the input cards. A channel may be input as one of the numbers if it is desired to use it in the equation. The detailed card format is shown in Tables 7 and 8.

AVG Cards (Cards No. 8 and 9, Figure 2)

This routine will create an average from any number of channels. The average is strictly linear, and there are no provisions for weight averaging. Any number of cards can be used to create the average. If seven or less channels are to be averaged, the averaging can be accomplished on one card (see Table 9). If more than seven channels are required, every eight new channels require a new card of the format shown in Table 10.

STOP Card (Card No. 10, Figure 2)

The stop card is a program control card that must be included at the conclusion of all AVG cards (see Table 11 for format).

Main Program Plot Cards (Cards No. 11 through 19, Figure 2)

These cards organize the program output for writing on disk storage for plotting. A series of nine cards (cards 11 through 19, Figure 2) is required for this program organization. Details for the plotting features are described in Section III.B.1.a of the main-text writeup, and the format for each card is shown in Table 12.

Appendix A

TABLE 7

ZNEW (PARAMETER) CARD

<u>Columns</u>	<u>Format</u>	<u>Description</u>
1 - 4	ZNEW	Generalized equation card used for creating new parameter.
5 - 12	Alphameric	New parameter name.
13	Blank	-
14 - 21	Alphameric or blank	First channel used in computation.
22 - 29	Alphameric or blank	Second channel used in computation.
30 - 37	Alphameric or blank	Third channel used in computation.
38 - 45	Alphameric or blank	Fourth channel used in computation.
46 - 53	Alphameric or blank	Fifth channel used in computation.
54 - 61	Alphameric or blank	Sixth channel used in computation.
62 - 69	Alphameric or blank	Seventh channel used in computation.
70 - 71	Blank	-
72	0	Code for addition, multiplication, division, or power equation.
	1	Code for fifth-order polynomial.
73 - 80	Blank	-

Appendix A

TABLE 8

ZNEW (CONSTANT) CARD

<u>Column</u>	<u>Format</u>	<u>Description</u>	
1 - 11	decimal	$\Delta 1$	
12 - 22	decimal	$\Delta 2$	
23 - 33	decimal	$\Delta 3$	Constants to be used in generalized equations.
34 - 44	decimal	$\Delta 4$	See Section III.B.3.d of main-text writeup for
45 - 55	decimal	$\Delta 5$	description of equations.
56 - 66	decimal	$\Delta 6$	
67 - 77	decimal	$\Delta 7$	
78 - 80	blank	-	

Appendix A

TABLE 9

AVG CARD (FIRST FORMAT)

There are two formats for this card. The following description applies to the first format.

<u>Columns</u>	<u>Format</u>	<u>Description</u>
1 - 3	AVG	For averaging different channels that measure same parameter.
4	Blank	
5 - 12	Alphameric	Name of parameter being averaged.
13	Blank	
14 - 15	Integer, right adjusted	Total number of channels being averaged.
16	Blank	
17 - 24	Alphameric	Channel number of first channel to be averaged.
25	Blank	
26 - 33	Alphameric	Channel number of second channel to be averaged.
34	Blank	
35 - 42	Alphameric	Channel number of third channel to be averaged (if none, leave blank)
43	Blank	
44 - 51	Alphameric	Channel number of fourth channel to be averaged (if none, leave blank)
52	Blank	
53 - 60	Alphameric	Channel number of fifth channel to be averaged (if none, leave blank).
61	Blank	
62 - 69	Alphameric	Channel number of sixth channel to be averaged (if none, leave blank).
70	Blank	
71 - 78	Alphameric	Channel number of seventh channel to be averaged (if none, leave blank).
79 - 80	(99 if no more channels are to be averaged; 1 if more channels are to be averaged.)	

The second format of this card is used only if there is a 1 in column 80 of the first or second format card; that is, if more channels are to be averaged than have been entered on the preceding AVG card or cards.

Appendix A

TABLE 10

AVG CARD (SECOND FORMAT)

<u>Columns</u>	<u>Format</u>	<u>Description</u>
1 - 3	AVG	
4 - 5	Blank	
6 - 13	Alphameric	Channel number of eighth channel to be averaged (if none, leave blank).
14	Blank	
15 - 22	Alphameric	Channel number of ninth channel to be averaged (if none, leave blank).
23	Blank	
24 - 31	Alphameric	Channel number of tenth channel to be averaged (if none, leave blank).
32	Blank	
33 - 40	Alphameric	Channel number of eleventh channel to be averaged (if none, leave blank).
41	Blank	
42 - 49	Alphameric	Channel number of twelfth channel to be averaged (if none, leave blank).
50	Blank	
51 - 58	Alphameric	Channel number of thirteenth channel to be averaged (if none, leave blank).
59	Blank	
60 - 67	Alphameric	Channel number of fourteenth channel to be averaged (if none, leave blank).
68	Blank	
69 - 76	Alphameric	Channel number of fifteenth channel to be averaged (if none, leave blank).
77 - 78	Blank	
79 - 80	(99 if no more channels are to be averaged; 1 if more channels are to be averaged.)	

Appendix A

TABLE 11

STOP CARD

<u>Column</u>	<u>Format</u>	<u>Description</u>
1 - 4	STØP	Required Control Card
5 - 80	Blank	-

Appendix A

TABLE 12

MAIN PROGRAM PLOT CARDS

P-2 Card

<u>Columns</u>	<u>Format</u>	<u>Description</u>
1 - 5	Blank	
6	Integer	Number of time intervals (1, 2, 3, or 4).
7 - 72	Blank	
73 - 80		

Repeat Cards P-3 through P-10* for each new time interval.

P-3 Card

<u>Columns</u>	<u>Format</u>	<u>Description</u>
1 - 12	Integer	CRT start time in tenths of a second (minimum: one-tenth of a second).
13 - 24	Integer	CRT stop time in tenths of a second (minimum: one-tenth of a second).
25 - 80	Blank	

P-4 Card

<u>Columns</u>	<u>Format</u>	<u>Description</u>
1 - 6	Integer, right-adjusted	Number of multiplots (<u>≤</u> 500).
7 - 80	Blank	

Repeat cards P-5 through P-10 for each new multiplot.

* Plot cards P-3 through P-10 are included in the main program and written on disk in order to save computer time that would be required to search through all 1100 channels on the NTO tape to find the required channels.

Appendix A

TABLE 12 (cont.)

P-5 Card

<u>Columns</u>	<u>Format</u>	<u>Description</u>
1 - 68	Alphameric	Multiplot title for Frame 1
69 - 72	Alphameric	Part of Multiplot title, used as multiplot code for Frame 2.
73 - 80	Blank	

P-6 Card

<u>Columns</u>	<u>Format</u>	<u>Description</u>
1 - 5	Blank	
6	Integer	Number (1, 2, or 3) of vertical scales per multiplot.
7 - 11	Blank	
12	(0 - if data are unclassified.) (1 - if data are classified.)	
13 - 80	Blank	

Repeat cards P-7 through P-10 for each new vertical scale.

P-7 Card

<u>Columns</u>	<u>Format</u>	<u>Description</u>
1 - 12	Decimal	Minimum value of specified vertical scale.
13 - 24	Decimal	Maximum value of specified vertical scale.
25 - 28	Blank	
29 - 30	Integer, right-adjusted	Number (≤ 10) of groups using scale.
31 - 80	Blank	

Appendix A

TABLE 12 (cont.)

Repeat cards P-8 through P-10 for each new group title.

P-8 Card

<u>Columns</u>	<u>Format</u>	<u>Description</u>
1 - 4	Blank	
5 - 6	Integer	Number (≤ 32) of channels in first group.
7 - 80	Blank	

Repeat card P-9 until all the channels in the group have been supplied.

P-9 Card

<u>Columns</u>	<u>Format</u>	<u>Description</u>
1 - 8	Alphameric	Number of first channel in the group.
13 - 20	Alphameric	Number of second channel in the group.
25 - 32	Alphameric	Number of third channel in the group.
37 - 44	Alphameric	Number of fourth channel in the group.
49 - 56	Alphameric	Number of fifth channel in the group.
61 - 68	Alphameric	Number of sixth channel in the group.
69 - 80	Blank	

P-10 Card

<u>Columns</u>	<u>Format</u>	<u>Description</u>
1 - 16	16 alphameric characters (including blanks)	First line of group title.
17 - 32	16 alphameric characters (including blanks)	Second line of group title.
33 - 80	Blank	-

Appendix A

A significant point for program input is the organization of the cards in this section. A misplaced card could destroy the entire plot formation in the Phase-I version of the program, and care must be taken to organize the cards in the proper sequence. The card format is shown in Table 12, and the following items summarize the sequencing of the cards:

(1) A P-2 card will initiate this section. This card indicates the number of time intervals to be plotted and is included only once.

(2) A P-3 card is placed next. This card indicates the time interval for a period to be plotted. All information to prepare the plots for this time interval will follow the P-3 card. After all information for plotting a given time interval is completed, a second P-3 (time-interval) card is inserted and is followed by its plot instructions. This procedure is repeated until the number of time intervals on the P-2 card is satisfied.

(3) Following each P-3 card, there is one P-4 card that indicates the number of multiplots in the time interval.

(4) The P-5 card that follows defines the title for an individual multiplot. There will be one P-5 card for each plot, and all information for an individual plot will follow the P-5 card. After a plot has been defined, the second P-5 card will be placed, and this routine will be repeated until all plots are defined.

(5) The P-6 card defines the number of scales per plot, and one P-6 card follows each P-5 card.

(6) The P-7 card defines one vertical scale. There will be one P-7 card for each scale on the plot. The P-7 cards will not follow each other. All data to be plotted on a given scale must be defined before definition of a second scale is initiated.

Appendix A

(7) The P-8 through P-10 cards define a group of channels. These cards will always follow each other in sequence and will be repeated until each group of channels on a given scale has been defined.

Override Cards (Block No. 5, Figure 1)

The override cards are identical in format to the card they are replacing in the master deck or main program. Not all routines can be changed with override cards, and the writeups on each routine must be reviewed to determine if override cards can be used.

Plot Routine (Block No. 6, Figure 1)

The plotting routine controls the removal of data from the disks and formation of the tape that drives a cathode-ray-tube plotter. A block diagram for inputting this portion of the program is shown in Figure 3. The detailed card input format for each card is shown in Table 13.

Cards P-1 and P-11 give information that will be common to all plots and are used to identify them with a specific test. There is one of each of these cards. These cards are not repeated. Cards P-12 and P-13 are used for the channel index of data plotted. The program user must cross-check these cards with the channels in the master deck to ensure that all channels will be properly indexed. There is one P-12 card but there is a P-13 card for each channel.

The P-3 through P-10 cards are identical to the P-3 through P-10 cards in the main program, and care should be taken to keep these identical. The repeated deck is used if it is desired to run the plot routine as a second step after writing the data on the disks with the main program.

Appendix A

Figure 3 - Card Deck Setup for Plot Routine

Appendix A

TABLE 13

PLOTTING PROGRAM

P-1 Card

<u>Columns</u>	<u>Format</u>	<u>Description</u>
1 - 72	Alphanumeric	Title for plot identification frame.
73 - 80	Blank	

P-11 Card

<u>Columns</u>	<u>Format</u>	<u>Description</u>
1 - 8	Alphanumeric	EP number.
9 - 16	Two integers followed by slash, repeated, and followed by two integers	Month, day, year of run.
17 - 80	Blank	

P-12 Card

<u>Columns</u>	<u>Format</u>	<u>Description</u>
1 - 2	Blank	
3 - 6	Integer, right-adjusted (up to 1100)	Total number of channels for which channel numbers are given (all channels plotted should be given).
7 - 80	Blank	

Appendix A

TABLE 13 (cont.)

P-13 Card

<u>Columns</u>	<u>Format</u>	<u>Description</u>
1 - 8	Alphameric	Channel number.
9 - 10	Blank	
11 - 65	Alphameric	Channel title.
66 - 80	Blank	

P-3 through P-10 Cards: These are identical to the P-3 through P-10 cards used for a time interval in the main program.

Appendix A

Steady-State Routine (Block No. 7, Figure 1)

This routine calls for the accumulated steady-state data from disks and prints it out in the desired format. The cards used to input this routine are shown in Table 14.

The master card deck is repeated for this routine, allowing the routine to be run in conjunction with, or separately from, the main program. An END card is required after the master deck for program control. The system-request card (Card No. 3, Table 14) allows the suppression of printing of portions of the data if they are not desired. Card No. 4 specifies the time intervals associated therewith. (A steady-state summary will be obtained for each time interval.) The systems requested and the time intervals requested must agree with information written on disk by the main program. The last card (Card No. 5, Table 14) is a control card that terminates the program.

Appendix A

TABLE 14
DATA-CARD FORMAT

<u>Card Type</u>	<u>Card Column</u>	<u>Format</u>	<u>Description</u>
1. NRO master deck			Identical to cards in master card deck.
2. Master-deck end card	1 - 3	END	
3. System-request card	1 - 2	Integer, right-adjusted	Up to a maximum of 17 requested numbers.
	3 - 4		
	5 - 6		
	7 - 8		
	.		
	.		
	.		
	.		
	33 - 34		
	70	Integer	Nonzero entry provides disk or tape data dump.
	73 - 80	Integer	Test date (e.g., 08/14/67).
4. Time-interval card	1 - 8	Integer, right-adjusted	Control-room time expressed in tenths of a second for the start of a desired time interval followed by its respective end time. Enter up to five sets of start and end times
	9 - 16		
	17 - 24		
	25 - 32		
	33 - 40		
	41 - 48		
	49 - 56		
	57 - 64		
	65 - 72		
	73 - 80		
5. Data-end card	1 - 2	1*	

APPENDIX B

FLOW DIAGRAM AND FORTRAN SOURCE-PROGRAM LISTING, X-ENGINE DATA PROGRAM

A flow diagram summarizing the sequence of program operation is shown in Figure 1, sheet 1. A complete FORTRAN source-program listing for the data program is included as Figure 2.

The program will normally be operated in two steps, a matching check to verify the pretest planning and a data run complete through all calculation plot preparation and steady-state summaries. The matching check is shown sequentially as steps A-1 through A-6 in Figure 1. This routine essentially compares the NRO input tape with the NTO master card deck for channel discrepancies. A printout of discrepancies is obtained (step A-4, Figure 1).

The main program is initiated by reading in the program control cards (step M-1) and the data processing instructions, steps M-2 through M-6. The program then makes a check to determine if all equations in the flow subroutine (see Section III.B.3.c of main-writeup) can be computed with the channels obtained from the NTO deck. An on-line printout is obtained (M-8) if any of the flows cannot be computed. The program will run with missing flows, but this information allows the monitoring engineer to decide if alternate instructions should be input into the program.

The next set of cards to be read into the computer is the M-9 or plotting subroutine cards. These cards direct specific channel data to be stored on disks for plotting purposes. The format and purpose of these cards are described in Appendix A.

Once the cards are read into the program, the conversion subroutine is called to read the NTO test tapes. These are described in Appendix C. The NTO engineering-unit tapes have been processed on a CDC 3200 unit that is not compatible with the IBM 360-65. Hence, a conversion program is necessary for language compatibility reasons (steps M-10 through M-11).

Appendix B

When one NTO test-data tape (subsystem) is used, the conversion program reads one frame of data at a time and compares the CRT time with the program input start time. All data prior to this time are rejected, whereas all data between the input start and stop times flow through the program normally, as will be explained. If two tapes (subsystems) are used, the program reads the first frame of each tape and checks the time (M-10). The tape with the lower time value is then read until the times from both tapes are identical. Thereafter, both tapes are read and the frame data are merged and rejected until the input start time is reached, after which the data are retained (M-11).

Each frame of data is merged and corrected from CDC 3200 to IBM 360-65 language, and the tie between word location and alphameric number of the channel is completed (M-12). At this point, the flow of data through the program is continuous. At all times, seven complete sets of data are retained in core storage, the eighth set being stored on tape and disks as a new set is brought in. The seven sets of data are necessary for the smoothing routine (see Section III.B.3.a of main writeup).

The first phase in processing the data is through the smoothing routine (M-13). In general, the smoothing routine will smooth the data from any channel or channels within any time period. The first three points from the smoothing routine are carried as raw data. Beyond this point, the program discards the raw data and retains only the smoothed points. Thus, a plot of both smoothed and raw data from any individual channel is nonobtainable. This limitation is caused by the maximum number of channels of data processed (1100), including program-created channels. The 1100 figure was arrived at by storage considerations.

The next step in the processing of data is the time-point averaging calculation (M-14). Multiple measurements of the same parameter can be arithmetically averaged in accordance with input instructions (see Section III.B.3.b of the main writeup).

Appendix B

Step M-15 involves the calculation of flow within the engine, various facilities systems, the duct, the steam generators, and other areas of interest. These flow calculations are coded in the program but can be changed by means of the override cards (see Section III.B.3.c of the main writeup). A hydrogen-properties subroutine is available to determine any desired properties as necessary to evaluate a specific flow.

Step M-16 is a special calculation process whereby a number of computations can be accomplished without recompiling the program. This step creates a new channel of data that can be carried along in the general flow of data in a manner similar to that for any other channel data. The two added general equations are shown in Section III.B.3.d of the main writeup

Channel averages over a time period for steady-state summaries are processed at step M-17. A time-point average channel can be averaged over a time span as well as an individual channel. The arithmetic mean of any channel, as well as the standard deviation, can be calculated at this point. These steady-state data (maximum of five per run) are stored on disk and tape (see Section III.B.1.b of the main writeup).

As noted at M-18, seven frames of data plus the steady-state running totals for the arithmetic mean and standard deviation are maintained in core. As a new frame of data is converted, M-19, the eighth frame, is stored on tape; and the steady-state summary data are stored on disk and tape (M-21)

As the data reach this point in processing, the plotting subroutine is called (M-23). The time associated with the eighth point data set is checked with the input time plotting interval (M-24). If the data set time is within that interval, the data are accepted; if not, they are rejected. All data accepted are stored on disk as a time-oriented data set (M-25). The subroutine continues to accept each set of data as long as the eighth data set time falls within the plotting time interval (M-26). The program is capable of handling

Appendix B

a maximum of four plotting-time intervals with a maximum of 500 time sets with 1100 individual channel plots per interval. Only those parameters called for in the plotting routine are stored. When all data have been processed through the program, the main plotting routine is called (M-27).

Briefly, the plotting routine takes the data stored on disks and rearranges it sequentially relative to time on an individual channel basis. The data are scaled and placed on a plotting tape with other pertinent information, such as channel names, plot titles, and plot format. The plotting-tape format is compatible with the input requirements of the Benson-Lerner cathode-ray-tube plotter.

The plotting routine must first rearrange the data previously stored on disks from an all-channel-data, time-set orientation to an individual-channel-data-set, time-sequenced orientation. The latter orientation is necessary because the channels are plotted one at a time. To facilitate the rearrangement of data, the plotting routine creates five temporary storage areas on disks (M-28). The routine then reads 80 time sets of data, previously stored on disks, into core (M-29). This array (80 time units by 1100 channels) is rearranged in core so that the data for each individual channel form an array, 80 time units by 1 channel by 1100 arrays (M-30). The first 220 channel arrays (80 time units by 1 channel) are then stored in the first temporary disk storage area (M-31); similarly, channel arrays 221 through 440, 441 through 660, 661 through 880, and 881 through 1100 are stored in temporary storage areas 2, 3, 4, and 5, respectively (M-32). This process is repeated with the next 80 time sets of data and is continued until all 500 time sets are processed. At this point, the first temporary storage area contains the first 220 individual channel arrays for all 500 time points. The other temporary disk storage areas similarly contain complete individual channel arrays in successive groups of 220 channels each (M-33). The arrays for each individual channel are read from the temporary storage areas and stored on disk sequentially as a single array for each channel (M-34 through M-35). The rearranging process briefly outlined above is repeated for each time interval not exceeding four.

Appendix B

The plotting-program input card deck provides for each multiplot scale or set of scales, as well as the channels to be plotted and the sequence of the multiplots. With this information, the plotting routine locates the first channel data to be plotted and processes the information to form Frame 1 of the plot. Frame 1 contains the channel names, plot scales, and multiplot title. It also processes the Frame-2 data that form the plotting grid (M-36 through M-37). The next steps consist of reading the first set of data and locating the second set on the disk (M-38 through M-39). The first set of data is then scaled over the whole time interval (M-41). All information related to the formation of the frames and scaled plot data is feed to a plotting tape (M-43). The process starting at (M-38) is repeated until the data for all channels required to form a multiplot are placed on the plotting tape (M-42). The process starting at M-36 is repeated for each multiplot until all four time intervals are covered. The program then prepares two plot indexes that summarize the curves that have been plotted (step M-44 and Section III.B.1.a of the main-text writeup).

The final step is the printout of the steady-state data that have been accumulated on the disk (step M-45). After the steady-state printout, the computer can be returned to other service; but the steady-state and plot information on the disk is not accessible to other programs. This information can be recalled by the data program at a later date if desired.

Figure 1 - Flow Chart for Program Operations
(Sheet 1 of 9)

Figure 1 - Flow Chart for Program Operations
(Sheet 2 of 9)

Figure 1 - Flow Chart for Program Operations
(Sheet 3 of 9)

Figure 1 - Flow Chart for Program Operations
(Sheet 4 of 9)

Figure 1 - Flow Chart for Program Operations
(Sheet 5 of 9)

Figure 1 - Flow Chart for Program Operations
(Sheet 6 of 9)

Figure 1 - Flow Chart for Program Operations
(Sheet 7 of 9)

Figure 1 - Flow Chart for Program Operations
(Sheet 8 of 9)

Figure 1 - Flow Chart for Program Operations
(Sheet 9 of 9)

Appendix B

Figure 2 - Fortran Service-Program Listings
(Pages 81 Through 167)

Appendix B

MAIN PROGRAM

1 -

1 of 87

8080 LISTING

COUNT

```

//DS7101 EXFC HCOMPIL,
//      PARAM.COMPILE='NAME=MAIN,HCD,MAP,DECK,LINAD,IP=2'
//COMPIL.SYSIN DD *
*****57101
C REON TEST DATA ANALYSIS PROGRAM *E57101* 57101
*****57101
C SECTION 1. READING INPUT CARDS 57101
C IMPLICIT REAL (A-H, O-Z) , INTEGER (I-N) 57101
C REAL * 8 H(2),C(2),IDDAY/8H /,ZNAME(1100) 57101
C FIX INTEGER CONSTANTS USE BY JOB E57101 57101
C DATA ISLUC2,MAXAV,MAXFUN/996,250,960/ 57101
C DIMENSION ITT(5),IDXT(5),IST(10),IAVGT(10),ITEM(125) 57101
C DIMENSION LUCAT(1100),LDATA(1100),NAME(2700) 57101
C EQUIVALENCE (ZNAME(1),NAME(1)) 57101
C DIMENSION IP(INT(20),LPOINT(25) 57101
C DATA IPOINT /'A...BR30CF00EN80NCR1PT10PT30PT60SR00SR01.001E11TE5057101
C 1TE68TE83TT10TT5099999999999/' 57101
C 1. LPOINT/25*0/,NONAME/' /,NUMID/' /,IFINI/'STOP'/ 57101
C 2. CHECKING & SETTING UP REQUIRED TABLES (FLOWS,AVERAGING)57101
C DIMENSION IDELP(6),IPREQ(30),ITREQ(30),IMISC(24) 57101
C IPREQ ARRAY CONTAINS REQUIRED PRESSURE CHANNEL NAMES *****57101
C DATA IPREQ/'PT172.A PT814.A PT226.A AV JCHAMPT883.A PT885.A AV P1057101
C 1RIAV PRFINAV PPMPOFT405W..FFT405X..FFT405Y..FFT409W..FFT409X..FFT409Y.F57101
C ITREQ ARRAY CONTAINS REQUIRED TEMPERATURE CHANNEL NAMES *****57101
C 2'/, ITREQ/'AV TTNINAV TTURVIT227.A AV TCHAMTE846.A TE847.A AV TTU57101
C 3RIAV TRFIN TT478..FFT478..FFT478..FFT479..FFT479..FFT479..F57101
C IMISC ARRAY CONTAINS PUMP INLET,DUCT, AND STREAM GENERATOR FLOW 57101
C CHANNEL NAMES ( CAN BE PRESSURE CHANNEL NAME )*****57101
C 4'/, IMISC/'FE014.R FT054..FFT055..FFT056..FFT057..FFT058..FFT059.57101
C 5.FFT060..FFT406W..FFT406X..FFT406Y.F /, 57101
C IDELP ARRAY CONTAINS PRESSURE DIFFERENCE CHANNEL NAMES FOR FLOWS 57101
C VENTURI FLOW EQUATIONS *****57101
C 6 IDELP/'PD163.A PD804.A PD209.A /' 57101
C DIMENSION LDP(3),LUCP(15),LOCT(15),IDLUC(17) 57101
C DIMENSION NAMFLO(42),IDLUCI(32) 57101
C NERVA ENGINE CALCULATED FLOW DATA NAMES **FIXED CANNOT BE CHANGE**57101
C DATA NAMFLO/'W PUMPINW PMPINW NOZ W NOZ1 W TURB1W TURB11W TUR57101
C 1B12W TURB13W TURB14W C DIL W C DIL1W TURB1W TURB11W HE CL W BECL157101
C 2 W BFCL2 W BLT CLW PUMP OW HBP W HBP1 POWER /' 57101
C FACILITY DUCT AND STEAM GENERATORS FLOW DATA NAMES 57101
C W = WATER,F = FUEL,D = LIQUID OXYGEN IN STEAM GENERATOR NAMES 57101
C DATA IDLUCI/'W DUC1 1W DUC11AW DUC1 2W DUC12AW DUC1 3W DUC13AW ELJ57101
C 1I 1W SG W 1W SG W 2W SG W 3W SG F 1W SG F 2W SG F 3W SG O 1W SG O 57101
C 22W SG O 3'/ 57101
*****57101
C 3. CHECKING FLOW CALCULATION 57101
C DIMENSION ZCONS(7,10),IZTAB(70),IZFLG(10) 57101
C DATA ZCONS/70*1.0/,IZTAB/70*0/,IZFLG/10*0/ 57101
C 4. CHECKING FLOW CODE OPTION 57101
C DIMENSION LFTAB(9),LFCODF(6) 57101
C 5. READING TEST DATA TAPF 57101

```

MAIN PROGRAM

2 -

2 of 87

R080 LISTING

COUNT

```

        DIMENSION NRTTD(2),NFILED(8),NGTIME(4)
        DATA NFILED/8*0/
        DIMENSION TIME(13,320),LEVENT(13,320),IDESCP(7,320),ILOC(320)
        EQUIVALENCE (TIME(1,1),LEVENT(1,1)),(ILOC(1),NGDATA(1))
C      6. SEVEN POINT SMOOTHING
        DIMENSION DATA(960,7),DATA1(960),DATA2(960),DATA3(960),DATA4(960)
1,      DATA5(960),DATA6(960),DATA7(960),TDATA(1992),WDATA(1100)
2,      IEVNT1(13),IEVNT2(13)
        EQUIVALENCE (ITIME1,TDATA(1)),(ITIME2,TDATA(997))
        EQUIVALENCE (IEVNT1(1),TDATA(981)),(IEVNT2(1),TDATA(1977))
        EQUIVALENCE (DATA1(1),DATA(1,1)),(DATA2(1),DATA(1,2)),
1      (DATA3(1),DATA(1,3)),(DATA4(1),DATA(1,4)),
2      (DATA5(1),DATA(1,5)),(DATA6(1),DATA(1,6)),
3      (DATA7(1),DATA(1,7))
        DIMENSION CF(7),CY(7)
        DATA CF/-2.0,3.0,6.0,7.0,6.0,3.0,-2.0/
C      7. AVERAGING MULTIPLE CHANNELS FUNCTIONS
        DIMENSION AVSIG(225),TEM(250)
C      7A CALCULATING PROPERTIES (ENTHALPY,DENSITY)
        DIMENSION ENTHPY(9),RHU(9)
C      8. CALCULATING FLOW RATES
        DIMENSION FLK(9),FLKW(28)
        DATA FLKW/1060.0,1060.0,1405.0,1405.0,1290.0,1290.0,198.5,3.33,
1 3.33,3.33,0.0166,0.03125,17.961,0.0166,0.03125,17.961,0.0166,
2 0.03125,17.961,0.0276,0.03666,17.4320,0.0276,0.03666,17.4320,
3 0.0276,0.03666,17.4320/
C      9. STEADY STATE AVERAGING
        DIMENSION SDATA(1100),SDNU(1100)
        DIMENSION NI(2),NDAY(3),NDATE(3)
        DATA NDAY,NDATE/' ',' '
        INTEGER*2 IFLAG(960),NAVTAB(250),NAVLOC(1000)
        INTEGER*2 NSTALY(960)
        INTEGER*2 NGDATA(1992)
        INTEGER*2 IBEST(40),NGTALY(960)
        COMMON LFLOC,LDP
1,      LFCODE,LFTAB,LOCP,LUCT,FLK,ENTHPY,RHU,WDATA
C      9A WRITING DISK DATA,TAPE DATA, PRINTING.
C*****
C      E57101 FORMAT STATEMENTS *** INPUT AND OUTPUT ***FOLLOWS
C*****
C      TOP OF THE PAGE HEADFR (TFST SERIES,DATE,TIME,PAGE NO.)
1 1 FORMAT (1H1,49X,12HTEST SERIES ,2A8,42X,4HPAGE,1X,12 / 50X,12HDATE
1 1 OF TEST,2X,2A8 / 50X,12HTIME OF TEST,2X,1A4)
2 2 FORMAT (1H ,44X,44HS U M M A R Y   D A T A   A N O M A L I E S )
C      SUBSYSTEM DATA (RECORDS READ,TAPE ERRORS,TIME ERRORS)
3 3 FORMAT (1H ,21X,26HS U B - S Y S T E M ( 1 ) ,38X,26HS U B - S Y S
1 T E M ( 2 ) /20X,7HRECORDS,10X,4HTAPE,12X,4HTIME,27X,7HRECORDS,10
2X,4HTAPE,12X,4HTIME /22X,4HREAD,11X,6HERRORS,10X,6HERRORS,27X,4HRE
3AD,11X,6HERRORS,10X,6HERRORS / 20X,18,8X,18,8X,18,24X,18,8X,18,8X,
418 )
C      HEADER FOR TEST DATA CHANNELS PRINTOUT
4 4 FORMAT (1H ,19X,8H* TEST *,6X,10H* SMOOTH *,5X,15H* STATISTICAL *,57101

```

MAIN PROGRAM

3 -

3 of 87

8080 LISTING

COUNT

```

120X,8H* TFST *,6X,10H* SMOOTH *,5X,15H* STATISTICAL * / 5X,4HDATA 57101
211X,6HP0INTS,6X,4HFLAG,3X,6HP0INTS,4X,4HFLAG,4X,6HP0INTS,6X,4HDATA57101
3,11X,6HP0INTS,6X,4HFLAG,3X,6HP0INTS,4X,4HFLAG,4X,6HP0INTS / 3X,8HC57101
4CHANNELS,10X,4HLOST,12X,9HDISCARDED,9X,9HDISCARDED,3X,8HCHANNELS,1057101
5X,4HLOST,12X,9HDISCARDED,9X,9HDISCARDED /) 57101
C PRINTOUT OF DATA CHANNELS AND DATA CHANNELS TALLIES 57101
5 FORMAT (1H ,2X,2A4,9X,17,6X,12,3X,17,4X,12,5X,17,4X,2A4,9X,17,6X,157101
12,3X,17,4X,12,5X,17) 57101
C BOTTOM OF PAGE HEADER 57101
6 FORMAT (/ 1H ,4X,A8,3HDAY,2X,4A4,75X,12HWO.N(),=0914 ) 57101
C ERROR(INPUT) MESSAGES 57101
90 FORMAT (1H0,T32,24HINPUT DATA ERROR,TYPE = ,1A4,2X,4HSYS=,12,2X,6H57101
1TOTAL=, 15) 57101
91 FORMAT (1H0,T32,39HRTT ERROR-READING INPUT, LAST CARD READ=,1A4,2X,57101
16HTOTAL=, 15 ) 57101
92 FORMAT (1H ,T32,39HEOF ERROR-READING INPUT, LAST CARD READ=,1A4,2X,57101
16HTOTAL=, 15 ) 57101
93 FORMAT (1H ,T32,2A4,2X,14,3X,12,3X,12 )
C ***** 57101
C ***** 57101
C ***** 57101
C SECTION 1. READING INPUT CARDS 57101
C ***** 57101
DO 99 I = 1,15
LOC(I) = 0
LOCT(I) = 0
IDLOC(I) = 0
IF ( I .GT. 3 ) GO TO 99
LOC(I) = 0
99 CONTINUE
READ (5,11) C(1),C(2),R(1),R(2),ITIME,NOCORR,NOTAP1,NZPLOT,N0EVNT 57101
11 FORMAT ( 2A8,1X,2A8,1X,1A4,14,13,13,13) 57101
DO 100 J = 1,960
NGTALY(J) = 0
NSTALY(J) = 0
LOCAT(J) = 0
DATA(J) = 0
100 CONTINUE
C READ TAPE CARD 57101
ASSIGN 5100 TO NRTYPE 57101
ASSIGN 5300 TO NREAD1 57101
READ (5,101,ERR=109,FND=999) NOMID,ITAPE,(ITT(J),IDXT(J),J = 1,5) 57101
101 FORMAT (1A4,1X,12,1X,5(16,1X,14)) 57101
NOCARD = NOCARD + 1
NODT = 1
IF ( ITAPE .EQ. 0 ) GO TO 110
IF ( ITAPE .EQ. 11 ) GO TO 110
ASSIGN 5050 TO NREAD1
ASSIGN 5050 TO NRTYPE
GO TO 110
109 WRITE (6,91) NOMID,NOCARD 57101
NRTT = NRTT + 1 57101

```

MAIN PROGRAM

4 -

4 of 87

8080 LISTING

COUNT

```

C      READ SMOOTHING(START STOP TIME) CARD          57101
110 READ (5,111,ERR=119,FND=999) NOMID, (IST(J), J = 1,10 ) 57101
111 FORMAT (1A4,2X,10(I6) )                          57101
      NOCARD = NOCARD + 1                             57101
      GO TO 120                                       57101
119 WRITE (6,91) NOMID,NOCARD                        57101
      NRIT = NRIT + 1                                57101
C      READ STEADY STATE (START,STOP TIME ) CARD      57101
120 READ ( 5,111,FRR=129,FND=999) NOMID,(IAVGT(J), J = 1,10 ) 57101
C      FORMAT SAME AS SMOOTHING INPUT CARD            57101
      NOCARD = NOCARD + 1                             57101
      ASSIGN 9010 TO NUSSAV                          57101
      GO TO 130                                       57101
129 WRITE (6,91) NOMID,NOCARD                        57101
      NRIT = NRIT + 1                                57101
C      READING FLOW CONSTANT INPUT CARDS              57101
130 I = 1                                             57101
140 READ (5,151,ERR=190,FND=999) NOMID,NUCN,TFM(1),TFM(2),TFM(3), 57101
      1 (ITEM(J), J = 1,6),IEND                      57101
151 FORMAT (1A4,1X,I2,1X,3F10.5,3(2X,2A4),2X,I2)      57101
      NOCARD = NOCARD + 1                             57101
      IF ( NOCN .EQ. 0 ) GO TO 160                    57101
      IF ( NOCN .GF. 20 ) GO TO 170                    57101
      IF ( NOCN .GT. 15 ) GO TO 160                    57101
      IF ( NOCN .GF. 10 ) GO TO 165                    57101
      IF ( TFM(1) .EQ. 0.0 ) GO TO 152                 57101
      FLK(NUCN) = TFM(1)                             57101
152 IF ( ITEM(1) .EQ. NONAME ) GO TO 153                57101
      IPRFQ(2*NUCN-1) = ITEM(1)                       57101
      IPRFQ(2*NUCN) = ITEM(2)                         57101
153 IF ( ITEM(3) .EQ. NONAME ) GO TO 155                57101
      ITREQ(2*NUCN-1) = ITEM(3)                       57101
      ITREQ(2*NUCN) = ITEM(4)                         57101
      IF ( NOCN .GT. 3 ) GO TO 160                    57101
155 IF ( ITEM(5) .EQ. NONAME ) GO TO 160                57101
      IDELP(2*NUCN-1) = ITEM(5)                       57101
      IDELP(2*NUCN) = ITEM(6)                         57101
160 IF ( IFND .GT. 98 ) GO TO 200                      57101
      IF ( NOCN .EQ. 0 ) GO TO 180                    57101
      GO TO 199                                       57101
165 J = NOCN - 9                                       57101
      I = ( 3 * J ) - 1                              57101
      DO 169 J = 1,3                                  57101
      IF ( TFM(J) .EQ. 0.0 ) GO TO 169                 57101
      FLKW(I+J+8) = TFM(J)                           57101
169 CONTINUE                                          57101
      GO TO 152                                       57101
170 J = NOCN - 19                                       57101
      IF ( J .EQ. 0 ) GO TO 175                      57101
      IF ( TFM(1) .EQ. 0.0 ) GO TO 175                57101
      FLKW(J) = TFM(1)                               57101
175 IF ( ITEM(1) .EQ. NONAME ) GO TO 160              57101

```


MAIN PROGRAM

5 -

5 of 87

8080 LISTING

COUNT

```

      IMISC(2*J-1) = ITEM(1)          57101
      IMISC(2*J) = ITEM(2)           57101
      GO TO 160                       57101
180  WRITE (6,90) N0MID,N0SUB,N0CARD  57101
      GO TO 199                       57101
190  WRITE (6,91) N0MID,N0CARD        57101
      NRTT = NRTT + 1                 57101
199  I = I + 1                        57101
      GO TO 140                       57101
200  N0FLOW = I                       57101
C                                     57101
      IEND = 0                        57101
C      N0CORR = 0 NO CORRECTION TO NTO INPUT DECK 57101
C      N0CORR = 1 FEW NTO CORRECTION TO BE MADE  57101
C      N0CORR .GT. 1 ALL NTO INPUT DECK TO BE READ 57101
      IF ( N0CORR .LF. 1 ) GO TO 201  57101
      N0FUNC = N0CORR + 1             57101
      GO TO 205                       57101
201  READ (10) N0CARD,(LPOINT(J), J = 1,20) 57101
      N0FUNC = LPOINT(20)             57101
      I = LPOINT(20)                  57101
      N0 = 2 * N0FUNC                 57101
      READ (10) N0CARD,(NAMF(J), J = 1,N0) 57101
      READ (10) N0CARD,(LOCAT(J), J = 1,N0FUNC) 57101
      DO 202 J = 1,N0FUNC             57101
      LDATA(J) = LOCAT(J)             57101
202  CONTINUE                         57101
      READ (10) N0CARD,(LOCAT(J), J = 1,N0FUNC) 57101
      DO 204 J = 1,N0FUNC             57101
      IFLAG(J) = LOCAT(J)             57101
      LOCAT(J) = J + 3                57101
204  CONTINUE                         57101
      READ(10,END=205)                57101
205  I = 1                           57101
      J = 1                           57101
      JI = 1                          57101
      NFIND = 0                       57101
      ITEM(4) = 0                     57101
      ITEM(5) = 0                     57101
C      TEST FOR TYPE OF INPUT TEST FUNCTION CARD 57101
210  DO 299 N0 = 1,MAXFUN             57101
      IF ( N0CORR .GT. 1 ) NFIND = 0  57101
      IF ( I .GE. N0FUNC ) GO TO 300  57101
      IF ( N0CORR .NE. 0 ) GO TO 211  57101
      GO TO 240                       57101
211  IF ( NFIND .NE. 0 ) GO TO 250    57101
      READ (5,212,ERR=280,END=300) ITEM(1),ITEM(2),ITEM(3),ITEM(6), 57101
      1 ITEM(7),ITEM(8),ITEM(4),IEND,N0SUB 57101
212  FORMAT (2A4,I3,T9,I1,1A1,1A1,57X,I2,I2,I1)
C      NOT FOUND FLAG                 57101
      NFIND = 1                       57101
      IF ( ITEM(1) .NE. N0NAME ) GO TO 213 57101

```

MAIN PROGRAM

6 -

6 of 87

8080 LISTING

COUNT

```

      NUMID = ITEM(1)                                57101
C     TEST NAME = BLANK      INPUT ERROR              57101
      GO TO 255                                        57101
213  IF ( ITEM(8) .NE. NONAME ) GO TO 214             57101
      ITEM(3) = ITEM(3) / 10                          57101
214  IF ( ITEM(7) .NE. NONAME ) GO TO 215             57101
      ITEM(3) = ITEM(3) / 10                          57101
215  NOCARD = NOCARD + 1                             57101
      IF ( NOCORR .LE. 1 ) GO TO 250                 57101
219  NAME(J) = ITEM(1)                               57101
      NAME(J+1) = ITEM(2)                             57101
      LDATA(I) = ITEM(3)                             57101
      LOCAT(I) = I + 3                               57101
      IFLAG(I) = ITEM(4)                             57101
      IF ( NOSUB .LE. 1 ) GO TO 240                 57101
220  LDATA(I) = LDATA(I) + ISLOC2                    57101
C     FIND POINTERS TO TEST NAME PARAMETERS TABLE  57101
240  IF ( NAME(J) .LT. IPPOINT(JI) ) GO TO 260        57101
      LPPOINT(JI) = J + 1                           57101
      JI = JI + 1                                    57101
      GO TO 240                                       57101
250  IF ( NAME(J) .LT. ITEM(1) ) GO TO 240            57101
      IF ( NAME(J+1) .LT. ITEM(2) ) GO TO 240        57101
      NFIND = 0                                       57101
      IF ( NAME(J) .GT. ITEM(1) ) GO TO 255          57101
      IF ( NAME(J+1) .GT. ITEM(2) ) GO TO 255        57101
      GO TO 219                                       57101
255  WRITE (6,90) NUMID,NOSUB,NOCARD                57101
      NFIND = 0                                       57101
      GO TO 299                                       57101
260  DO 261 L = 1,12                                57101
      IF ( NAME(J) .NE. IMISC(2*L-1) ) GO TO 261     57101
      IF ( NAME(J+1) .NE. IMISC(2*L) ) GO TO 261     57101
      IDLOC(L) = I                                   57101
      GO TO 290                                       57101
261  CONTINUE                                        57101
262  DO 265 L = 1,3                                  57101
      IF ( NAME(J) .NE. IDELP(2*L-1) ) GO TO 265     57101
      IF ( NAME(J+1) .NE. IDELP(2*L) ) GO TO 265     57101
      LDP(L) = I                                     57101
      GO TO 290                                       57101
265  CONTINUE                                        57101
266  DO 270 L = 1,15                                  57101
      IF ( NAME(J) .NE. IPREQ(2*L-1) ) GO TO 270     57101
      IF ( NAME(J+1) .NE. IPREQ(2*L) ) GO TO 270     57101
      LOCP(L) = I                                    57101
      GO TO 290                                       57101
270  CONTINUE                                        57101
271  DO 275 L = 1,15                                  57101
      IF ( NAME(J) .NE. ITREQ(2*L-1) ) GO TO 275     57101
      IF ( NAME(J+1) .NE. ITREQ(2*L) ) GO TO 275     57101
      LOCT(L) = I                                    57101

```

MAIN PROGRAM

7 -

7 of 87

8080 LISTING

COUNT

```

      GO TO 290
275 CONTINUE
      GO TO 290
280 WRITE (6,281)
281 FORMAT (T32,'RTT READING TEST FUNCTION CARD')
290 J = J + 2
      I = I + 1
      IF ( IEND .GT. 98 ) GO TO 300
299 CONTINUE
300 NOFUNC = I - 1
      J = J - 2
      IFUN = 2 * I
      IFLUC = IFLUC(1)
C     POINTER FOR AVERAGE TEST NAMES
      LPOINT(JI) = J + 3
C     CHECK AND STORE TEST FUNCTION TO BE AVERAGED
C     INITIALIZE
      IFLUC = J + 2
      K = IFLUC
      NULBCI = NOFUNC + 3
      L = 1
      DO 399 I = 1,MAXAV
        READ (5,301,FRR=320,FND=400) NOMID,NAMF(K),NAMF(K+1),
1          NAVTAB(I),(ITEM(J),J=1,14),IEND1
301 FORMAT (1A4,2A4,1X,12,7(1X,2A4),12)
      IF ( IEND1 .LE. 0 ) GO TO 310
      IF ( ITEM(13) .EQ. NUNAME ) GO TO 310
      IF ( IEND1 .GT. 98 ) GO TO 310
      NJ = 15
      J1 = 30
305 READ (5,302,FRR=320,FND=400) NOMID,( ITEM(J), J = NJ,J1 ),IEND1
302 FORMAT (1A4,8(1X,2A4),2X,12)
      IF ( IEND1 .LE. 0 ) GO TO 310
      IF ( ITEM(J1-1) .EQ. NUNAME ) GO TO 310
      IF ( IEND1 .GT. 98 ) GO TO 310
      J1 = J1 + 16
      NJ = NJ + 16
      GO TO 305
310 IF ( NOMID .EQ. IFINI ) GO TO 400
      J1 = 2 * NAVTAB(I)
      DO 314 J = 1,20
        IF ( J .EQ. 18 ) GO TO 340
        IF ( LPOINT(J) .GE. ITEM(1) ) GO TO 340
314 CONTINUE
315 WRITE (6,90) NOMID,NOSUB,NOCARD
      GO TO 380
320 WRITE (6,91) NOMID,NOCARD
      NRIT = NRIT + 1
      GO TO 380
340 JI = LPOINT(J-1)
      NO = JI
      M = J

```

MAIN PROGRAM

4 -

8 of 87

8080 LISTING

COUNT

```

      DO 350 J = M,20                                57101
      IF ( J .EQ. 18 ) GO TO 360                      57101
      IF ( IPDINT(J) .GT. ITEM(JI-1) ) GO TO 360      57101
350  CONTINUE                                         57101
      GO TO 315                                         57101
360  M = LPDINT(J)                                     57101
      DO 369 J = 1,J1,2                               57101
361  IF ( ITEM(J) .NE. NAME(JI-1) ) GO TO 365         57101
      IF ( ITEM(J+1) .NE. NAME(JI) ) GO TO 365       57101
      NAVLOC(L) = JI / 2                             57101
      GO TO 366                                         57101
365  JI = JI + 2                                       57101
      IF ( JI .LT. M ) GO TO 361                     57101
C      CANNOT FIND REQUESTED CHANNEL NAME ** PRINTOUT CHANNEL NAME 57101
      WRITE (6,1101) ITEM(J),ITEM(J+1)              57101
C      LOWER NO. OF AVERAGE DATA CHANNEL TO BE USE FOR BEST ESTIMATE 57101
      NAVTAB(I) = NAVTAB(I) - 1                      57101
      IF ( NAVTAB(I) .EQ. 0 ) GO TO 399              57101
      JI = NO                                         57101
      GO TO 369                                         57101
366  L = L + 1                                         57101
369  CONTINUE                                         57101
380  NOCARD = NOCARD + 1                             57101
      LDATA(I+NOFUNC) = I + NOFUNC                   57101
      LOCAT(I+NOFUNC) = I + NOLOC                     57101
      K = K + 2                                       57101
      IF ( IEND1 .GT. 98 ) GO TO 500                 57101
399  CONTINUE                                         57101
400  IF ( IEND1 .LT. 98 ) I = I - 1                  57101
500  NUAVG = I                                         57101
999  WRITE (6,92) NUMID,NOCARD                      57101
C*****57101
C      2. CHECKING & SETTING UP REQUIRED TABLES (FLOWS,AVERAGING)57101
C*****57101
C      COMPARE INPUT TEST FUNCTION NAME WITH PROGRAM'S REQUIRED LIST 57101
C      CHECK PRESSURE,TEMPERATURE,AND (FLOWS,DELTA P.,ETC) RESPECTIVELY 57101
1000 J1 = IFLOC                                       57101
      DO 1099 I1 = 1,NUAVG                             57101
      J = 1                                           57101
      DO 1020 I = 1,30,2                               57101
      IF ( LOCP(J) .NE. 0 ) GO TO 1020                57101
      IF ( NAME(J1) .NE. IPREQ(I) ) GO TO 1020        57101
      IF ( NAME(J1+1) .NE. IPREQ(I+1) ) GO TO 1020    57101
      LOCP(J) = (J1+1) / 2                           57101
      GO TO 1090                                       57101
1020 J = J + 1                                         57101
      J = 1                                           57101
      DO 1040 I = 1,30,2                               57101
      IF ( LOCT(J) .NE. 0 ) GO TO 1040                57101
      IF ( NAME(J1) .NE. ITREQ(I) ) GO TO 1040        57101
      IF ( NAME(J1+1) .NE. ITREQ(I+1) ) GO TO 1040    57101
      LOCT(J) = (J1+1) / 2                           57101

```

MAIN PROGRAM

9 -

9 of 87

8080 LISTING

COUNT

```

GO TO 1090 57101
1040 J = J + 1 57101
      J = 1 57101
      DO 1060 I = 1,24,2 57101
      IF ( IDLOC(J) .NE. 0 ) GO TO 1060 57101
      IF ( NAME(J1) .NE. IMISC(I) ) GO TO 1060 57101
      IF ( NAME(J1+1) .NE. IMISC(I+1) ) GO TO 1060 57101
      IDLOC(J) = (J1+1) / 2 57101
      GO TO 1090 57101
1060 J = J + 1 57101
      J = 1 57101
      DO 1080 I = 1,6,2 57101
      IF ( LDP(J) .NE. 0 ) GO TO 1080 57101
      IF ( NAME(J1) .NE. IDELP(I) ) GO TO 1080 57101
      IF ( NAME(J1+1) .NE. IDELP(I+1) ) GO TO 1080 57101
      LDP(J) = (J1+1) / 2 57101
      GO TO 1090 57101
1080 J = J + 1 57101
1090 J1 = J1 + 2 57101
1099 CONTINUE 57101
C CHECK TO SEE IF ALL REQUIRED TEST FUNCTION FOUND 57101
  I = 1 57101
1101 FORMAT (T32,2A4,2X,'COULD NOT BE FOUND ** CHECK FOR NAME CHANGE') 57101
      DO 1199 J = 1,15 57101
      IF ( LOCP(J) .NE. 0 ) GO TO 1150 57101
      WRITE (6,1101) IPREQ(I),IPREQ(I+1) 57101
1150 I = I + 2 57101
1199 CONTINUE 57101
      I = 1 57101
      DO 1299 J = 1,15 57101
      IF ( LOCT(J) .NE. 0 ) GO TO 1250 57101
      WRITE (6,1101) ITREQ(I),ITREQ(I+1) 57101
1250 I = I + 2 57101
1299 CONTINUE 57101
      I = 1 57101
      DO 1399 J = 1,11 57101
      IF ( IDLOC(J) .NE. 0 ) GO TO 1350 57101
      WRITE (6,1101) IMISC(I),IMISC(I+1) 57101
1350 I = I + 2 57101
1399 CONTINUE 57101
      I = 1 57101
      DO 1499 J = 1,3 57101
      IF ( LDP(J) .NE. 0 ) GO TO 1450 57101
      WRITE (6,1101) IDELP(I),IDELP(I+1) 57101
1450 I = I + 2 57101
1499 CONTINUE 57101
C POINTER FOR FLOWS AND POWER TEST NAMES 57101
  LPOINT(19) = K + 1 57101
C CHECKING FOR REQUIRED DUCT AND STEAM GENERATOR FLOW PARAMETERS ** 5710
  DO 2050 I = 10,15,3 57101
  IF ( LOCP(I) .EQ. 0 .OR. LOCT(I) .EQ. 0 ) GO TO 2050 57101
  IDLOC(I+2) = 1 57101

```

MAIN PROGRAM

10 -

10 of 87

H080 LISTING

COUNT

```

      IF ( L0CP(I+1) .EQ. 0 ) GO TO 2020
      IDLOC(I+3) = 1
      LOCT(I+1) = LOCT(I)
2020  IF ( L0CP(I+2) .EQ. 0 ) GO TO 2050
      IDLOC(I+4) = 1
      LOCT(I+2) = LOCT(I)
2050  CONTINUE
      M = NOAVG + NOFUNC + 1
      I8A = M - 1
      J1 = 1
      DO 2099 J = 1,16
      IF ( IDLOC(J+1) .EQ. 0 ) GO TO 2099
      NAME(K) = IDLOC1(2*J-1)
      NAME(K+1) = IDLOC1(2*J)
      LDATA(M) = M
      LOCAT(M) = M + 3
      M = M + 1
      J1 = J1 + 1
      K = K + 2
2099  CONTINUE
      I8B = J1 - 1
      N = M - 1
      NOLOC = N + 3
      I8C = N
      DO 2199 I = 1,21
      LDATA(I+N) = I + N
      LOCAT(I+N) = I + NOLOC
      NAME(K) = NAMELO(2*I-1)
      NAME(K+1) = NAMELO(2*I)
      K = K + 2
2199  CONTINUE
      N = N + 21
      NOSSP = N
C*****
C      READING INPUT CARDS TO CALCULATE NEW TEST DATA PARAMETER
C      USE IN ADDF SUBROUTINE DEPENDING ON TYPE OF EQUATION CODE
C*****
      IZ = 1
      IF ( NOMID .EQ. IFINI ) GO TO 2500
      I8D = N
      I = 0
2402  READ (5,2403,ERR=2409,END=2499) NOMID,NAME(K),NAME(K+1),
      1 (ITEM(I6),ITEM(I6+1), I6 = 1,14,2),IZFLG(IZ)
2403  FORMAT (3A4,1X,7(2A4),2X,I1)
      IF ( NOMID .EQ. IFINI ) GO TO 2500
      READ (5,2404,ERR=2409,END=2499) (TEM(I6), I6 = 1,7)
2404  FORMAT (7E11.5)
      DO 2408 I6 = 1,7
      IF ( ITEM(2*I6-1) .EQ. NONAME ) GO TO 2407
      DO 2406 L = 1,NOSSP
      IF ( ITEM(2*I6-1) .NE. NAME(2*L-1) ) GO TO 2406
      IF ( ITEM(2*I6) .NE. NAME(2*L) ) GO TO 2406

```

MAIN PROGRAM

11 -

11 of 87

8080 LISTING

COUNT

```

      I/IA*(I+16) = 1                      57101
      GO TO 2407                             57101
2406 C CONTINUE                             57101
C      CANNOT FIND REQUESTED CHANNEL NAME ** PRINTOUT CHANNEL NAME      57101
      WRITE (6,1101) ITEM(2*IB-1),ITEM(2*IB) 57101
2407 /CJIS(16,12) = TRN(16)                 57101
2408 C() T100                               57101
      J = ( K + 1 ) / 2                     57101
      LDATA(J) = N + I/                     57101
      LUCAT(J) = U + IZ + 3                 57101
      IZ = IZ + 1                           57101
      K = K + 1                             57101
      I = I + 1                             57101
      GO TO 2402                             57101
2409 GO TO 2402                             57101
2499 IF ( IZ .GT. 10 ) GO TO 2500           57101
      IZLC(IZ) = -1                         57101
2500 IZ = IZ - 1                           57101
      NDISP = NDISP + I/                   57101
      LPOINT(20) = K                       57101
      LPOINT(21) = NDISP                   57101
      LPOINT(22) = NDISP                   57101
      LPOINT(23) = NDISP                   57101
      K = K - 1                             57101
C      WRITE POINT KS FOR DIRECTORY RECORD      57101
      NOCARD = 1                           57101
      WRITE (13) NOCARD,LPOINT              57101
      WRITE (08) NOCARD,LPOINT              57101
C      WRITE TEST NAMES DIRECTORY RECORD        57101
      NOCARD = 2                           57101
      WRITE (13) NOCARD,( NAME(I), I = 1,K) 57101
      WRITE (08) NOCARD,( NAME(I), I = 1,K) 57101
C      *****XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX 57101
C      3. CHECKING FLOW CALCULATION            57101
C      *****XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX 57101
C      (CHECKING FOR REQUIRED TEST FUNCTION TO CALCULATE FLOW RATES 57101
      3(00) DO 01 I = 1,9                    7101
      IF TA4(I) = 0                          57101
      3(01) CONTINUE                         57101
C      IF ( IFLOW .GT. 0 ) GO TO 3100          57101
      LFCOEF(I) = 1                         57101
      3100 DO 3200 I3 = 1,9                  57101
      IF ( FLK(I3) .GT. 0.0 ) GO TO 3200      57101
      IF ( I3 .GT. 7 ) GO TO 3200            57101
      GO TO 3500                             57101
      3200 IF ( LDP(I3) .GT. 0 .OR. I3 .GT. 3 ) GO TO 3300 57101
      GO TO 3500                             57101
      3300 IF ( LUCP(I3) .GT. 0 ) GO TO 3400 57101
      GO TO 3500                             57101
      3400 IF ( LUCT(I3) .GT. 0 ) GO TO 3600 57101
      IF ( I3 .GT. 8 ) GO TO 3600           57101

```

MAIN PROGRAM

12 -

12 of 87

8080 LISTING

COUNT

```

3500 LFTAB(I3) = 1                                57101
3600 CONTINUE                                    57101
C*****57101
C      4. CHECKING FLOW CODE OPTION                57101
C*****57101
      CALL CKFLOW                                57101
C      WRITE DATA LOCATION RECORD FOR TEST NAMES  57101
      NOCARD = 3                                57101
      WRITE (13) NOCARD, ( LOCAT(I), I = 1,NOSSP) 57101
      WRITE (08) NOCARD, ( LOCAT(I), I = 1,NOSSP) 57101
C      CHECKING START AND STOP TIME FOR STEADY STATE AVERAGING 57101
      IEND = 1                                    57101
      IEND1 = 1                                    57101
      IAVT1 = IAVGT(IEND1)                        57101
      IAVT2 = IAVGT(IEND1+1)                      57101
      IF ( IAVT1 .EQ. 0 ) GO TO 4950                57101
      IF ( IAVT2 .EQ. 0 ) GO TO 4950                57101
      DO 4949 I9 = 1,NOSSP                         57101
      SDNU(I9) = 0.0                                57101
      SDATA(I9) = 0.0                                57101
4949 CONTINUE                                    57101
      IEND1 = IEND1 + 2                            57101
      GO TO 4970                                    57101
4950 ASSIGN 9500 TO NOSSAV                        57101
C      CHECKING START AND STOP TIME FOR SMOOTHING  57101
C*****57101
4970 IST1 = IST(IEND)                            57101
      IST2 = IST(IEND+1)                          57101
      IF ( IST1 .EQ. 0 ) GO TO 4980                57101
      IF ( IST2 .EQ. 0 ) GO TO 4980                57101
      ISMTHF = 1                                    57101
      IEND = IEND + 2                              57101
      GO TO 4990                                    57101
4980 ISMTHF = 0                                    57101
      GO TO 4990                                    57101
C*****57101
4990 ASSIGN 5230 TO NREAD                        57101
C      INITIALIZATION OF 7 POINT SMOOTHING SECTION 57101
      ITREC = 0                                    57101
      ASSIGN 6001 TO NGATE                        57101
      ASSIGN 6700 TO NGATE1                      57101
      ICYCLF = 864000                             57101
      ITWR = ITT(NUDT)                            57101
      CSIG = 3.00                                  57101
      NORFC2 = 1                                    57101
      NOREC1 = 1                                    57101
      NRTTD(1) = 0                                57101
      NRTTD(2) = 0                                57101
C      NOCARD = 0 TEST DATA RECORD TYPE          57101
      NOCARD = 0                                    57101
      NOCORR = 10                                  57101
C      IIFOR (INTEGER) = 4040 (BCD) ** END TAPE FLAG ** NTO TAPE 57101

```


MAIN PROGRAM

13 -

13 of 87

8080 LISTING

COUNT

```

C      I1EOR = 1212712                                57101
C      ASSIGN 5311 TO NCYCLE                            57101
C      IT2 = 0                                           57101
C      IQUIT = 0                                         57101
C      NDOOR1 = 0                                         57101
C      NDOOR2 = 0                                         57101
C      DO 4995 J = 1,960                                57101
C      NGTALY(J) = 0                                     57101
4995  CONTINUE                                           57101
C      DO 4999 J = 1,1992                                57101
C      NGDATA(J) = 0                                     57101
4999  CONTINUE                                           57101
C      INITIALIZATION FOR PLOT SECTION (NAME ARRAY,NUMBER) 57101
C      ENTER THIS SECTION ONLY ONCE                     57101
C      IF ( NZPLUT .NE. 0 ) GO TO 5000                  57101
C      CALL WNAME (ZNAME,NOSSP)                          57101
C*****                                                57101
C      5. READING TEST DATA TAPE                       57101
C*****                                                57101
5000  GO TO NRTYPE, (5050,5100)                          57101
5050  READ (10,ERR=5075,END=5080) TDATA                 57101
C      GO TO 5500                                         57101
5075  NRTTD(1) = NRTTD(1) + 1                            57101
C      GO TO 5050                                         57101
5080  REWIND 10                                           57101
C      GO TO 9990                                         57101
C      READING NTU( TEST DATA) TAPE                    57101
C      READ TAPE LABEL * STORING PERTINENT INFORMATION  57101
C      IOK = 0 , RECORD READ CORRECTLY                  57101
C      IOK = 4 , ENCOUNTER TAPE MARK,                   57101
C      OTHERWISE ,READING ERRUR *COUNT ERRORS*        57101
5100  NSUBSY = 1                                           57101
C      CALL JDHD1 (ITEM(1),IOK)                          57101
C      IF ( IOK .EQ. 0 ) GO TO 5105                      57101
C      IF ( IOK .EQ. 4 ) GO TO 5198                      57101
C      GO TO 5150                                         57101
5105  NOTEST = ITEM(1)                                    57101
C      NOTAPE = ITEM(3)                                   57101
C      NTTYPE = ITEM(4)                                   57101
C      IF ( NSUBSY .NE. ITEM(2) ) GO TO 5175             57101
C      IF ( NTTYPE .NE. 2 ) GO TO 5175                   57101
5110  NSUBSY = 2                                           57101
C      IF ( NDOOR1 .EQ. 1 ) GO TO 5200                   57101
C      IF ( ITAPE .EQ. 11 ) GO TO 5200                   57101
C      CALL JDHD2 (ITEM(5),IOK)                          57101
C      IF ( IOK .EQ. 0 ) GO TO 5115                      57101
C      IF ( IOK .EQ. 4 ) GO TO 5199                      57101
C      GO TO 5150                                         57101
5115  IF ( ITEM(8) .NE. 2 ) GO TO 5175                   57101
C      IF ( NOTAPE .NE. ITEM(7) ) GO TO 5175             57101
C      IF ( NDOOR2 .EQ. 1 ) GO TO 5215                   57101

```

MAIN PROGRAM

14 -

14 of 87

8080 LISTING

COUNT

```

      GO TO 5200
5130 CALL JDUNLD (NOTAPE,NSUBSY)
C      UNIT 9 = ONLINE PRINTER
      WRITE (9,5131) IT1
5131 FORMAT (1H ,12HLAST TIME = ,I6)
      IF ( NOTAPE .GE. NOTAP1 ) GO TO 5145
      GO TO 5100
5135 IQUIT = 1
      IF ( ISNTHF .NE. 0 ) GO TO 6151
      GO TO 7000
5140 CALL JDUNLD (NOTAPE,NSUBSY)
C      UNIT 9 = ONLINE PRINTER
      WRITE (9,5131) IT1A
      IF ( NOTAPE .GE. NOTAP1 ) GO TO 5135
      GO TO 5110
5150 NRTTD(NSUBSY) = NRTTD(NSUBSY) + 1
      WRITE (6,5151) NSUBSY
5151 FORMAT (1H ,T32,'RTT ERROR READING NT0 TAPE LABEL,SUB-SYSTEM = ',
1I2 )
      IF ( NSUBSY .LT. 2 ) GO TO 5110
      GO TO 5200
5175 WRITE (6,5176)
5176 FORMAT (' ',T32,'WRONG TYPE OF TAPE MOUNTED,NOTIFY CUSTOMER****')
      IF ( NSUBSY .LT. 2 ) GO TO 5110
      IF ( NDDOR2 .NE. 0 ) GO TO 5215
      GO TO 5200
5198 IIEUF = 1
      GO TO 5110
5199 I2FUF = 1
      GO TO 5200
5200 READ FILE LABEL
5200 NSUBSY = 1
      CALL JDHDI (ITEM(1),IOK)
      IF ( IOK .EQ. 0 ) GO TO 5201
      IF ( IOK .EQ. 4 ) GO TO 5298
      GO TO 5250
C      CHECK FOR CONTINUOUS FILE ON NEXT TAPE
C      ITEM(1) = IIEUF,END OF TAPE, WRITE MESSAGE TO OPERATOR
5201 IF ( ITEM(1) .EQ. IIFOR ) GO TO 5290
      IIFOF = 0
      IF ( ITFW(4) .EQ. 0 ) GO TO 5205
      IF ( NFILED(3) .NE. ITEM(3) ) GO TO 5205
      GO TO 5229
5205 DO 5210 I = 1,4
      NFILED(I) = ITEM(I)
5210 CONTINUE
      IF ( NDDOR1 .NE. 0 ) GO TO 5229
5215 NSUBSY = 2
      IF ( ITAPE .EQ. 11 ) GO TO 5216
      CALL JDHDP (ITEM(5),IOK)
      IF ( IOK .EQ. 0 ) GO TO 5216
      IF ( IOK .EQ. 4 ) GO TO 5299

```

MAIN PROGRAM

15 -

15 of 87

H080 LISTING

COUNT

```

      GO TO 5225
5216 IF ( ITEM(5) .EQ. IIFOR ) GO TO 5295
      I2FUF = 0
      IF ( ITAPE .EQ. 11 ) GO TO 5232
      IF ( ITEM(4) .EQ. 0 ) GO TO 5220
      IF ( NFILED(7) .NE. ITEM(7) ) GO TO 5220
      GO TO 5229
5220 DO 5225 I = 1,4
      NFILED(I+4) = ITEM(I+4)
5225 CONTINUE
5229 GO TO NREAD, (5230,5280,5281,5300,5305)
5230 IF ( NTAPE .EQ. 0 ) GO TO 5232
      IF ( NFILED(2) .NE. 5 ) GO TO 5275
      IF ( ITEM(6) .NE. 3 ) GO TO 5277
5232 ASSIGN 5300 TO NREAD
      GO TO 5300
C     RTT ERROR COUNT AND RETURN
5250 NRTTD(NSUBSY) = NRTTD(NSUBSY) + 1
      WRITE (6,5251) NSUBSY
5251 FORMAT (1H ,T32,'RTT ERROR READING NTD FILE LABEL,SUB-SYSTEM = ',
      1I2 )
      IF ( NSUBSY .LT. 2 .AND. ITAPE .NE. 11 ) GO TO 5215
      GO TO 5229
5275 SKIPPING FILE      NOT TEST DATA TYPE
5275 ASSIGN 5280 TO NREAD
      IF ( ITAPE .EQ. 11 ) GO TO 5280
5277 ASSIGN 5281 TO NREAD
      GO TO 5280
5280 CALL JDRC1 (TDATA(1),IOK)
      IF ( ITIME1 .EQ. IIFOR ) GO TO 5290
      IF ( IOK .EQ. 4 ) GO TO 5298
5281 IF ( ITAPE .EQ. 11 ) GO TO 5280
      CALL JDRC2 (TDATA(997),IOK)
      IF ( ITIME2 .EQ. IIFOR ) GO TO 5295
      IF ( IOK .EQ. 4 ) GO TO 5299
      GO TO 5280
5290 NDJUR1 = 1
      ASSIGN 5280 TO NREAD
      GO TO 5130
5295 NDJUR2 = 1
      ASSIGN 5281 TO NREAD
      GO TO 5140
5298 I1FUF = 1 + I1FUF
      IF ( I1FUF .GT. 1 ) GO TO 5135
      NDJUR1 = 0
      ASSIGN 5230 TO NREAD
      GO TO 5200
5299 I2FUF = 1 + I2FUF
      IF ( I2FUF .GT. 1 ) GO TO 5135
      NDJUR2 = 0
      ASSIGN 5230 TO NREAD
      GO TO 5215

```

16 - 16 of 87

COUNT

96

MAIN PROGRAM

17 -

17 of 87

8080 LISTING

COUNT

CALL JDCNVT (TDATA(994),2,NGCNT)	57101
GO TO 5311	57101
5315 IF (IT1 .GT. IT2) GO TO 5400	57101
5320 IDT = IT1 - IT2	57101
IF (IDT .LT. -838800) GO TO 5325	57101
GO TO 5330	57101
C ADDITIONAL CHECK FOR RECYCLING FLAG	57101
5325 IF (IT1 .LE. 10) GO TO 5340	57101
5330 NGTIME(3) = NGTIME(3) + 1	57101
GO TO 5300	57101
5340 ASSIGN 5312 TO NCYCLE	57101
GO TO 5312	57101
5350 NDOUR1 = 1	57101
ASSIGN 5300 TO NREAD	57101
GO TO 5130	57101
5360 NDOUR2 = 1	57101
ASSIGN 5305 TO NREAD	57101
GO TO 5140	57101
C HAD TIME TALLY FOR EACH SUB-SYSTEMS	57101
5370 NGTIME(1) = NGTIME(1) + 1	57101
DO 5371 I = 2,996	57101
TDATA(I) = -0.0	57101
5371 CONTINUE	57101
IT1A = IT1A	57101
NG = NG + 1	57101
GO TO 5314	57101
5372 NGTIME(2) = NGTIME(2) + 1	57101
DO 5373 I = 2,996	57101
TDATA(I+996) = -0.0	57101
5373 CONTINUE	57101
IT1A = IT1	57101
NG = NG + 1	57101
IF (NG .GE. 2) GO TO 5300	57101
GO TO 5400	57101
5375 NRTTD(NSUBSY) = NRTTD(NSUBSY) + 1	57101
IF (NSUBSY .LF. 1) GO TO 5300	57101
GO TO 5305	57101
5398 I1EOF = 1 + I1EOF	57101
NDOUR1 = 1	57101
ASSIGN 5300 TO NREAD	57101
GO TO 5200	57101
5399 I2EOF = 1 + I2EOF	57101
NDOUR2 = 1	57101
ASSIGN 5305 TO NREAD	57101
GO TO 5215	57101
C WRITING RAW ENGINEERING UNIT TAPE IN 360/DS FORMAT	57101
5400 IF (IDXT(NODT) .EQ. 0) GO TO 5450	57101
IF (IT1 .GE. ITWR) GO TO 5450	57101
GO TO 5300	57101
5425 ITWR = ITT(NODT+1)	57101
NODT = NODT + 1	57101
GO TO 5500	57101

MAIN PROGRAM

18 -

18 of 87

8080 LISTING

COUNT

```

5450 IF ( ITAPE .EQ. 11 ) GO TO 5500          57101
      WRITE (10) TDATA                        57101
      ITWR = IT1 + IDXT(NODT)                 57101
      IF ( IT1 .GT. ITT(NODT+1) ) GO TO 5425  57101
5500 IF ( NOEVNT .EQ. 0 ) GO TO 5700          57101
      NOEVNT = 1                             57101
5505 DO 5599 I = 1,13                        57101
C     CHECKING FAST EVENTS                   57101
      IF ( NSUBSY .EQ. 1 ) GO TO 5506         57101
      LDATA1 = IEVNT2(I)                     57101
      GO TO 5508                             57101
5506 LDATA1 = IEVNT1(I)                      57101
5508 CALL TESBIT (LDATA1,ITFM,NSUBSY)         57101
      IF ( I .GT. 10 ) GO TO 5550            57101
      DO 5549 J = 1,12                       57101
5510 IF ( ITEM(J) .EQ. 0 ) GO TO 5540         57101
      K = LEVENT(13,NOEVNT)                  57101
      IF ( K .GT.13 ) GO TO 5540              57101
      LEVENT(K,NOEVNT) = IT1                 57101
      LEVENT(13,NOEVNT) = LEVENT(13,NOEVNT) + 1 57101
5540 NOEVNT = NOEVNT + 1                     57101
      IF ( I .GT. 10 ) GO TO 5589            57101
5549 CONTINUE                               57101
5590 IF (NOEVNT .GT. MXEVNT ) GO TO 5600     57101
5599 CONTINUE                               57101
      GO TO 5600                             57101
C     CHECKING SLOW EVENTS                   57101
5550 DO 5589 J = 1,16                       57101
      GO TO 5510                             57101
5589 CONTINUE                               57101
      GO TO 5599                             57101
5600 IF ( NSUBSY .EQ. 2 ) GO TO 5700         57101
      NSUBSY = 2                             57101
      IF ( ITAPE .EQ. 11 ) GO TO 5700        57101
      GO TO 5505                             57101
C     RESTORE TEST DATA INTO WORKING STORAGE 57101
5700 DO 5799 I = 1,NDFUNC                    57101
      DATA1(I) = TDATA(LDATA(I))            57101
5799 CONTINUE                               57101
C*****57101
C      6. SEVEN POINT SMOOTHING              57101
C*****57101
C     SEVEN POINT SMOOTHING SECTION (7 TEST RECORDS IN STORAGE) 57101
C     NEWSUM = (-2.0Y1 + 3.0Y2 + 6.0Y3 + 7.0Y4 + 6.0Y5 + 3.0Y6 -2.0Y7) 57101
C     NEWY4 = NEWSUM / 21.0                  57101
C     NEWY4 = NEWSUM / 21.0                  57101
C     NEWY4 = NEWSUM / 21.0                  57101
6000 ITRNXT = IT1                           57101
      GO TO NGATE, (6001,6100,6150,7000)     57101
6001 IF ( ISMTHF .GT. 0 ) GO TO 6050         57101
      ASSIGN 7000 TO NGATE                    57101
      GO TO 6700                             57101

```

MAIN PROGRAM

19 -

19 of 87

8080 LISTING

COUNT

```

6050 IF ( IT1 .GE. IST1 ) GO TO 6060          57101
      IF ( ITREC .LE. 0 ) GO TO 6700          57101
      GO TO 7000                              57101
6060 ASSIGN 6100 TO NGATE                     57101
      ASSIGN 6196 TO NGATE2                   57101
      ASSIGN 6850 TO NGATE3                   57101
      ASSIGN 6910 TO NGATE4                   57101
      IT7 = 0                                57101
      IT6 = 0                                57101
      IT5 = 0                                57101
      IT4 = 0                                57101
      IT3 = 0                                57101
C      TEST (DATA TIME .VS. SMOOTHING TIME INTERVAL ) 57101
6100 IF ( ITREC .LE. 0 ) GO TO 6800           57101
      IF ( IT1 .LT. IST1 ) GO TO 7000        57101
      ASSIGN 6150 TO NGATE                   57101
      ASSIGN 6800 TO NGATE1                  57101
6150 IF ( IT1 .LE. IST2 ) GO TO 6195         57101
6151 ASSIGN 6900 TO NGATE1                   57101
      DO 6180 I6 = 1,NDFUNC                  57101
      WDATA(I6) = DATA4(I6)                57101
6180 CONTINUE                               57101
      GO TO 6600                             57101
6195 GO TO NGATE2, (6196,6200)              57101
6196 IF ( IT7 .GT. 0 ) GO TO 6199            57101
      IF ( IT5 .EQ. 0 ) GO TO 7000          57101
      GO TO 6800                             57101
6199 ASSIGN 6200 TO NGATE2                   57101
      ASSIGN 6859 TO NGATE3                  57101
C      STORING DATA FOR WRITING(CALC. FLOWS,.FTC.) AND SMOOTHING 57101
6200 DO 6599 I6 = 1,NDFUNC                  57101
      IF ( IFLAG(I6) .EQ. 1 ) GO TO 6201     57101
      IF ( IFLAG(I6) .NE. 11 ) GO TO 6599    57101
6201 SUMD = 0.0                             57101
      SUMD2 = 0.0                           57101
      SUMN = 0.0                             57101
      DO 6299 J = 1,7                        57101
      CY(J) = DATA(I6,J)                   57101
      IF ( CY(J) .LE. 0.0 ) GO TO 6299       57101
      SUMN = 1.0 + SUMN                     57101
      SUMD = CY(J) + SUMD                   57101
      SUMD2 = CY(J)**2 + SUMD2              57101
6299 CONTINUE                               57101
C      CALCULATE MEAN(XBAR),SIGMA,+NSIGMA,-NSIGMA 57101
6300 IF ( SUMN .LT. 4.0 ) GO TO 6400         57101
      XBAR = SUMD / SUMN                    57101
      SIGMA = ( (SUMD2-(SUMD/SUMN*SUMD)) / (SUMN-1.0) ) 57101
      SIGMA = SQRT (ABS(SIGMA))              57101
      SIG3 = CSIG * SIGMA                   57101
      SIG1 = XBAR + SIG3                    57101
      SIG2 = XBAR - SIG3                    57101
C      TEST EACH OF 7 POINTS,1ST INITIALIZE DIVISOR 57101

```

20 - 20 of 87

COUNT

[illegible]

H0H0 LISTING

C UNT

```

6919 CONTINUE                                57101
      ITRFC = IT3                            57101
      ITRNXT = IT2                          57101
      ASSIGN 6950 TO NGATE4                  57101
      GO TO 7000                             57101
6950 DO 6969 I = 1,NUFUNC                   57101
      WDATA(I) = DATA2(I)                  57101
6969 CONTINUE                                57101
      ITRFC = IT2                            57101
      ITRNXT = IT1                          57101
C      REINITIALIZE FOR NEXT SMOOTHING PERIOD 57101
      IF ( IST(IFND) .EQ. 0 ) GO TO 6979     57101
      IST1 = IST(IFND)                      57101
      IST2 = IST(IFND+1)                    57101
      IEND = IFND + 2                       57101
      IF ( IFND .LT. 10 ) GO TO 6980         57101
6979 ISMTHF = 0                             57101
6980 ASSIGN 6001 TO NGATEF                  57101
      ASSIGN 6700 TO NGATE1                  57101
C*****57101
C      7. AVERAGING MULTIPLE CHANNELS FUNCTIONS 57101
C*****57101
C      COMPUTE AVERAGE MULTI-CHANNELS TEST FUNCTION 57101
C      COMPUTE MEAN, SIGMA, TEST EACH DATA CHANNEL, RECOMPUTE NEW XBAR AND 57101
C      UNCERTAINTIES FOR BEST ESTIMATE      57101
7000 J1 = 1                                57101
      IF ( NDAVG .LE. 1 ) GO TO 7800         57101
      DO 7799 I7 = 1, NDAVG                 57101
      SUMD = 0.0                            57101
      SUMN = 0.0                            57101
      SUMD2 = 0.0                           57101
      J2 = 1                                57101
      M = NAVTAB(I7)                        57101
      DO 7399 J = 1,M                       57101
7300 TEM(J2) = WDATA(NAVLOC(J1))            57101
      IBEST(J2) = NAVLOC(J1)                57101
      SUMD = SUMD + TEM(J2)                 57101
      SUMN = SUMN + 1.0                     57101
      SUMD2 = TEM(J2)**2 + SUMD2            57101
7390 J2 = J2 + 1                           57101
      J1 = J1 + 1                           57101
7399 CONTINUE                              57101
      IFIRST = 0                            57101
C      COMPUTE XBAR, SIGMA                    57101
7400 IF ( SUMN .EQ. 0.0 ) GO TO 7790        57101
      XBAR = SUMD / SUMN                    57101
      SIG4 = XBAR * .001                    57101
      IF ( SUMN .EQ. 1.0 ) GO TO 7795       57101
      SIGMA = ( (SUMD2-(SUMD/SUMN*SUMD)) / (SUMN-1.0) ) 57101
7450 SIGMA = SQRT (ABS(SIGMA))              57101
      SIG3 = CSIG * SIGMA                  57101
      IF ( SIG3 .LE. SIG4 ) SIG3 = SIG4     57101

```

8080 LISTING

COUNT

```

      IF ( SUMN .LE. 2.0 ) GO TO 7795          57101
      SIG1 = XBAR + SIG3                      57101
      SIG2 = XBAR - SIG3                      57101
      IF ( IFIRST .NE. 0 ) GO TO 7795         57101
      SUMD = 0.0                              57101
      SUMN = 0.0                              57101
      SUMD2 = 0.0                             57101
      M = J2 - 1                             57101
      DO 7599 I = 1,M                         57101
      IF ( TEM(I) .GT. SIG1 ) GO TO 7550       57101
      IF ( TEM(I) .GE. SIG2 ) GO TO 7560       57101
7550  NGTALY( IBEST(I) ) = NGTALY( IREST(I) ) + 1 57101
      GO TO 7599                             57101
7560  SUMD = SUMD + TEM(I)                   57101
      SUMN = SUMN + 1.0                       57101
      SUMD2 = TEM(I)**2 + SUMD2               57101
7599  CONTINUE                              57101
C      COMPUTE NEW XBAR AND UNCERTAINTIES      57101
      IFIRST = 1                             57101
      GO TO 7400                             57101
7790  AVSIG(I7) = 0.0                       57101
      WDATA(I7+NOFUNC) = -0.0                57101
      GO TO 7799                             57101
7795  AVSIG(I7) = SIG3                       57101
      WDATA(I7+NOFUNC) = XBAR                 57101
7799  CONTINUE                              57101
C*****57101
C      7A CALCULATING PROPERTIES (ENTHALPY,DENSITY) 57101
C*****57101
C      COMPUTING PROPERTIES -ENTHALPY,DENSITY      57101
C      7800 DO 7850 I7= 1,8                  57101
      IF ( LFTAB(I7) .NE. 0 ) GO TO 7849       57101
      IF ( WDATA(LOCP(I7)) .LE. 14.7 ) GO TO 7849 57101
      PRESS = WDATA(LOCP(I7))                 57101
      IF ( WDATA(LOCT(I7)) .LE. 36.0 ) GO TO 7849 57101
      TEMP = WDATA(LOCT(I7))                  57101
      CALL TABTP (TEMP,PRESS,H,Z,GAMMA,CSURP,SV,S,TK,VIS,ICE) 57101
      IF ( ICE .LT. 0 ) GO TO 7849             57101
      ENTHPY(I7) = H                          57101
      RHO(I7) = (Z*TEMP*9200.0) / (PRESS*1728.0) 57101
      RHO(I7) = 1.0 / RHO(I7)                 57101
      GO TO 7850                              57101
7849  ENTHPY(I7) = -0.0                      57101
      RHO(I7) = -0.0                          57101
7850  CONTINUE                              57101
C      INITIALIZE FLOW DATA STORAGES            57101
      DO 7899 I = I8A,NOSSP,1                 57101
      WDATA(I+1) = -0.0                       57101
7899  CONTINUE                              57101
C*****57101
C      8. CALCULATING FLOW RATES                 57101

```

MAIN PROGRAM

23 -

23 of 87

8080 LISTING

COUNT

```

C*****57101
C   COMPUTING DUCT FLOWS AND STEAM GENERATOR FLOWS (H2O,LUX,FUEL) 57101
C   DUCT FLOWS (GAL / MIN) STEAM GENERATOR FLOWS (LB / SEC)      57101
C       J = 1                                                    57101
C       IF ( I8B .EQ. 0 ) GO TO 8499                             57101
C       DO 8099 I = 2,11                                         57101
C       IF ( IDLOC(I) .EQ. 0 ) GO TO 8099                         57101
C       TEM(1) = WDATA(IDLOC(I))                                 57101
C       IF ( TEM(1) .LE. 0.0 ) GO TO 8099                        57101
C       TEM(2) = SQRT (TEM(1))                                   57101
C       WDATA(J+I8A) = TEM(2) * FLKW(I-1)                       57101
8098 J = J + 1                                                    57101
C       IF ( J .GT. I8B ) GO TO 8499                             57101
8099 CONTINUE                                                    57101
C       DO 8199 I = 1,6                                           57101
C       IF ( LUCP(I+9) .EQ. 0 ) GO TO 8199                       57101
C       IF ( LUCT(I+9) .EQ. 0 ) GO TO 8199                       57101
C       TEM(1) = WDATA(LUCP(I+9))                                57101
C       IF ( TEM(1) .LE. 70.0 ) GO TO 8198                       57101
C       TEM(2) = WDATA(LUCT(I+9))                                57101
C       WDATA(J+I8A) = ( (TEM(1) * FLKW(3*I+8))-(TEM(2) * FLKW(3*I+9)) + 57101
C       1 FLKW(3*I+10) )                                         57101
8198 J = J + 1                                                    57101
C       IF ( J .GT. I8B ) GO TO 8499                             57101
8199 CONTINUE                                                    57101
8499 CALL FLOWS                                                  57101
C   RESTORE CALCULATED FLOWS INTO WRITING STORAGES              57101
C   DO 8599 I = 1,21                                              57101
C   WDATA(I+I8C) = WDATA(I+I079)                                57101
8599 CONTINUE                                                    57101
C   COMPUTING NEW DATA BASED ON TYPE OF EQUATION (CODE)        57101
C   USING ADDF SUBROUTINE                                         57101
C   IF ( IZ .EQ. 0 ) GO TO 9000                                  57101
C   I7 = 0                                                         57101
C   DO 8799 I = 1,IZ                                              57101
C   NDEQ = IZFLG(I)                                              57101
C   IF ( NDEQ .LT. 0 ) GO TO 8799                                57101
C   DO 8699 J = 1,7                                                57101
C   J1 = I7 + J                                                  57101
C   IF ( IZTAB(J1) .EQ. 0 ) GO TO 8695                           57101
C   TEM(J) = WDATA(IZTAB(J1)) * ZCONS(J,I)                      57101
C   GO TO 8699                                                    57101
8695 TEM(J) = ZCONS(J,I)                                         57101
8699 CONTINUE                                                    57101
C   CALL ADDF (ANS,TEM(1),NDEQ)                                  57101
C   WDATA (I+I8D) = ANS                                           57101
C   I7 = 7 * I                                                    57101
8799 CONTINUE                                                    57101
C*****57101
C   9. STEADY STATE AVERAGING                                   57101
C*****57101
C   57101

```

MAIN PROGRAM

24 -

24 of 87

8080 LISTING

COUNT

```

C      STEADY STATE AVERAGING FROM IAVT1(SEC) TO IAVT2(SEC)          57101
C                                                                 57101
9000 GO TO NUSSAV, (9010,9500)          57101
9010 IF ( ITRNXT .LT. IAVT1 ) GO TO 9500          57101
      IF ( ITRNXT .GT. IAVT2 ) GO TO 9100          57101
C      STORE DATA FOR STEADY STATE PERIOD          57101
      DO 9099 I9 = 1,NUSSP          57101
      IF ( WDATA(I9) .LE. 0.0 ) GO TO 9099          57101
      SDATA(I9) = WDATA(I9) + SDATA(I9)          57101
      SDNU(I9) = 1.0 + SDNU(I9)          57101
9099 CONTINUE          57101
      GO TO 9500          57101
9100 DO 9199 I9 = 1,NUSSP          57101
      IF ( SDNU(I9) .LE. 0.0 ) GO TO 9199          57101
      SDATA(I9) = SDATA(I9) / SDNU(I9)          57101
9199 CONTINUE          57101
C      WRITE STEADY STATE DATA          57101
      WRITE (08) NOCURR,IAVT1,IAVT2,(SDATA(I9), I9 = 1,NUSSP)          57101
      WRITE (13) NOCURR,IAVT1,IAVT2,(SDATA(I9), I9 = 1,NUSSP)          57101
C      CHECKING START AND STOP TIME FOR STEADY STATE AVERAGING          57101
      IAVT1 = IAVGT(IEND1)          57101
      IAVT2 = IAVGT(IEND1+1)          57101
      IF ( IAVT1 .EQ. 0 ) GO TO 9300          57101
      IF ( IAVT2 .EQ. 0 ) GO TO 9300          57101
      IEND1 = IEND1 + 2          57101
      DO 9299 I9 = 1,NUSSP          57101
      SDNU(I9) = 0.0          57101
      SDATA(I9) = 0.0          57101
9299 CONTINUE          57101
      GO TO 9500          57101
9300 ASSIGN 9500 TO NUSSAV          57101
C*****          57101
C      9A WHITING DISK DATA,TAPE DATA, PRINTING.          57101
C*****          57101
9500 WRITE (13) NOCURR,ITREC,ITRNXT,(WDATA(J), J = 1,NUSSP)          57101
9501 IF ( IQUIT .EQ. 1 ) GO TO 9980          57101
      IF ( NZPLOT .NE. 0 ) GO TO 9950          57101
C      PROCESSING TEST DATA RECORD *** TIME= ITRNXT,DATA = WDATA          57101
      CALL WDISK (ITREC,WDATA)          57101
9950 GO TO NGATE1, (6700,6800,6900)          57101
9980 IF ( ITAPE .NE. 11 .OR. ITAPE .NE. 0 ) GO TO 9990          57101
      FND FILE 10          57101
      REWIND 10          57101
9990 END FILE 13          57101
      REWIND 13          57101
C      LAST TIME ON TAPE *** SIGNAL END TO PLOTTING SUBROUTINE          57101
      IF ( NZPLOT .NE. 0 ) GO TO 9991          57101
      CALL #MAX (IT1)          57101
9991 CALL ASJATE (NDAY)          57101
      CALL IDAY (NDAY,NI)          57101
      CALL GRGRDA (NI(2),1,NI(1),IDDAY)          57101
      CALL ASPATE (NDATF)          57101

```

MAIN PROGRAM

25 -

25 of 87

8080 LISTING

COUNT

```

CALL IDATE (NDATE) 57101
DO 9993 J = 1,6 57101
K = J - 1 57101
CALL AGCHED (K,B,C) 57101
9993 CONTINUE 57101
C PRINT ALL DATA CHANNELS BEING USED 57101
M = 1 57101
N = 50 57101
J = 1 57101
I1 = 50 57101
J1 = 100 57101
LINE = 0 57101
IF ( NOFUNC .GE. 100 ) GO TO 9995 57101
I1 = (NOFUNC+1) / 2 57101
J1 = 2 * I1 57101
C PRINT OUT TOP OF PAGE HEADER 57101
9995 LINE = 1 + LINE 57101
WRITE (6,1) C,LINE,B,ITIME 57101
WRITE (6,2) 57101
WRITE (6,3) NOREC1,NRTTD(1),NGTIME(1),NORFC2,NRTTD(2),NGTIME(2) 57101
WRITE (6,4) 57101
C PRINTOUT DATA 57101
DO 9996 I = M,N,1 57101
ITEM(1) = NGDATA(LDATA(I)) 57101
ITEM(2) = NGDATA(LDATA(I+I1)) 57101
WRITE (6,5) NAME(J),NAME(J+1),ITFM(1), 57101
1 IFLAG(I),NSTALY(I),IFLAG(I),NGTALY(I), 57101
2 NAME(J+J1),NAME(J+1+J1),ITEM(2), 57101
3 IFLAG(I+I1),NSTALY(I+I1),IFLAG(I+I1),NGTALY(I+I1) 57101
J = J + 2 57101
IF ( (I+I1) .GE. NOFUNC ) GO TO 9997 57101
9996 CONTINUE 57101
WRITE (6,6) IDDAY,NDATE 57101
M = M + 100 57101
IF ( M .GT. NOFUNC ) GO TO 9998 57101
N = N + 100 57101
J = J + J1 57101
IF ( (M+99) .LE. NOFUNC ) GO TO 9995 57101
I = NOFUNC - M 57101
I = (I+1) / 2 57101
I1 = I 57101
J1 = 2 * I1 57101
GO TO 9995 57101
9997 WRITE (6,6) IDDAY,NDATE 57101
C***** 57101
C CHECK FOR TEST EVENT OPTION ***** 57101
9998 IF ( NOEVNT .EQ. 0 ) GO TO 1699 57101
C READING TEST EVENT DESCRIPTION INPUT CARDS 57101
NO = 1
1605 READ (5,1601,END=1610) ZNAME(NO),ILUC(NO),(IDESC(J,NO), J = 1,7) 57101
1601 FORMAT (1A8,I3,7A4)
NO = NO + 1

```

MAIN PROGRAM

26 -

26 of 87

8080 LISTING

COUNT

```

      GD TO 1605
1610 NO = NO - 1
      DO 1630 I = 1,NO
      NOEVNT = ILOC(I)
C      CONVERT INTEGER CRT TIMES TO FLOATING CRT TIMES
      DO 1629 J = 1,13
      TIME(J,NOEVNT) = LEVENT(J,NOEVNT)
      IF ( J .GT. 12 ) GO TO 1629
      TIME(J,NOEVNT) = TIME(J,NOEVNT) / 10.0
1629 CONTINUE
1630 CONTINUE
C      PRINT TEST EVENT AND TEST EVENT (CRT) TIMES
      IT1 = ITT(1)
      CALL WEVENT(NO,ZNAME,ILOC,IDESCP,TIME,B,C,IT1,IT2)
1699 CONTINUE
C*****
      CALL CDEX (100)
9999 STOP
      END

/*
      SUBROUTINE WEVENT (NO,ZNAME,ILOC,IDESCP,TIME,B,C,IT1,IT2)
C      SUBROUTINE FOR TEST EVENT OUTPUT PRINTOUT
      IMPLICIT REAL(A-H,O-Z), INTEGER(I-N)
      DIMENSION TIME(13,320),IDESCP(7,320),ILOC(1100),LOCT1(16),LOCT(12)
1,      ITEM(9),IHD(3)
      DATA LOCT/3,2,1,6,5,4,9,8,7,12,11,10/,
1      LOCT1/4,3,2,1,8,7,6,5,12,11,10,9,16,15,14,13/
      REAL*8 B(2),C(2),ZNAME(1100),HED(3)
      DATA IHD/' ON OFF NO '/,HED/' OPEN CLOSED EVENTS '/
C      NO = NUMBER OF EVENT INPUT CARDS READ
C      ZNAME = ARRAY CONTAINING CHANNEL NO.
C      ILOC = ARRAY CONTAINING THE EVENT NUMBER
C      IDESCP = ARRAY CONTAINING THE EVENT DESCRIPTION
C      LEVENT = ARRAY CONTAINING THE TEST EVENT (CRT) TIMES
C      C,B = ARRAY CONTAINING THE TEST SERIES AND TEST DATE
C      IT1 AND IT2 = START CRT TIME AND STOP CRT TIME
C      FORMAT STATEMENTS FOR TEST EVENT PRINTOUT
1      FORMAT (1H1,120X,7HPAGE NO,14 / 60X,13HX-ENGINE TEST,30X,12HTEST S57101
1      SERIES ,2A8/ 61X,11HEVENTS DATA,31X,12HTEST DATE ,2A8 / 47X,18HC057101
2      NTR0L ROOM TIME ,F8.1,4H TO ,F8.1 //)
2      FORMAT (1H ,8H CHANNEL,3X,12HTEST EVENT ,5X,10H SWITCH(S) 10HSTART57101
1      ON ,3X,1A4,6X,1A4,6X,1A4,6X,1A4,6X,1A4,6X,1A4,6X,1A4,6X,1A4 /1X,57101
2      H NUMBERS,3X,12HDESCRIPTIONS,5X,10H VALVE(V) ,10HSTART OPEN,1X, 57101
3      1A8,2X,1A8,2X,1A8,2X,1A8,2X,1A8,2X,1A8,2X,1A8,2X,1A8,2X,1A8 /)
3      FORMAT ( 1H ,1A8,2X,7A4,9(1X,F8.1,1X))
4      FORMAT (1H ,8H CHANNEL,3X,12HTEST EVENT ,5X,10H SWITCH(S) 10H *****57101
1      **** ,3X,1A4,6X,1A4,6X,1A4,6X,1A4 /1X,57101
2      H NUMBERS,3X,12HDESCRIPTIONS,5X,10H VALVE(V) ,10H ***** ,1X, 57101
3      1A8,2X,1A8,2X,1A8,2X,1A8 /)
5      FORMAT (1H ,1A8,2X,7A4,10X,3(1X,F8.1,1X),F9.0)
6      FORMAT (/ 4X,9A4,78X,12HWD.NO.= 0914 )
      CALL GRYDAY(ITEM)

```

MAIN PROGRAM

27 -

27 of 87

8080 LISTING

COUNT

```

      TIME1 = IT1                      57101
      TIME1 = TIME1 / 10.0             57101
      TIME2 = IT2                      57101
      TIME2 = TIME2 / 10.0             57101
      ITYPE = 0                       57101
      NOPAGE = 1                      57101
C 11 LINE = 70                        57101
      PRINTOUT CHANNEL NO.(EVENT ID),TEST EVENT DESCRIPTION,CRT TIMES 57101
      DO 60 I = 1,N0                  57101
      IF ( ILUC(I) .EQ. 0 ) GO TO 60    57101
      NOEVNT = ILUC(I)                 57101
      LOCSUB = 0                       57101
      IF ( NOEVNT .GT. 120 ) GO TO 15   57101
C 12 LUC0 = NOEVNT / 13                57101
      LUC1 = LUC0 * 12                 57101
      LUC2 = NOEVNT - LUC1             57101
      LUC0 = LOCT(LUC2) + LUC1         57101
      GO TO 19                         57101
C 15 IF ( NOEVNT .LE. 160 ) GO TO 16   57101
      LOCSUB = 160                    57101
      NOEVNT = NOEVNT - LOCSUB         57101
      IF ( NOEVNT .LE. 280 ) GO TO 12   57101
C 16 LUC0 = NOEVNT / 17                57101
      LUC1 = LUC0 * 16                 57101
      LUC2 = NOEVNT - LUC1             57101
      LUC0 = LOCT1(LUC2) + LUC1        57101
C 19 LOCATE = LUC0 + LOCSUB            57101
      NOEVNT = ILUC(I)                 57101
      IF ( ITYPE .NE. 0 ) GO TO 30     57101
C 20 IF ( LINE .LT. 70 ) GO TO 21      57101
      TOP OF PAGE HEADER PRINTOUT     57101
      WRITE (6,1) NOPAGE,C(1),C(2),B(1),B(2),TIME1,TIME2             57101
      WRITE (6,2) IHD(2),IHD(1),IHD(2),IHD(1),IHD(2),IHD(1),IHD(2), 57101
      1 IHD(1),HED(2),HED(1),HED(2),HED(1),HED(2),HED(1),HED(1), 57101
      2 HED(2),HED(1)                 57101
      LINE = 1                         57101
C 21 WRITE (6,3) ZNAME(I),(IDFSCP(K,NOEVNT), K = 1,7), 57101
      1 (TIME(J,LOCATE), J = 1,9)      57101
      GO TO 50                         57101
C 30 IF ( LINE .LT. 70 ) GO TO 40      57101
      WRITE (6,1) NOPAGE,C(1),C(2),B(1),B(2),TIME1,TIME2             57101
      WRITE (6,4) IHD(2),IHD(1),IHD(2),IHD(3),HED(2),HED(1),HED(2), 57101
      1 HED(3)                         57101
      LINE = 1                         57101
C 40 WRITE (6,5) ZNAME(I),(IDFSCP(K,NOEVNT), K = 1,7), 57101
      1 (TIME(J,LOCATE), J = 10,13)    57101
C 50 LINE = LINE + 1                   57101
      IF ( LINE .LT. 70 ) GO TO 60     57101
      WRITE (6,6) (ITEM(J), J = 1,9)   57101
C BOTTOM OF PAGE HEADER PRINTOUT      57101
      NOPAGE = NOPAGE + 1              57101
C 60 CONTINUE                         57101

```

MAIN PROGRAM

28 -

28 of 87

FORTRAN LISTING

COUNT

```

      IF ( LINE .GE. 70 ) GO TO 65
      WRITE (6,6) (ITEM(J), J = 1,9)
      NUPAGE = NUPAGE + 1
65  IF ( ITYPE .NE. 0 ) GO TO 70
      ITYPE = 1
      GO TO 11
70  RETURN
      END
      SUBROUTINE ADDF(FNEW,A,NPOLY)
C     NEW DATA GENERAL EQUATION ,IF NDFQ CDDF = 0
C     FNEW = ((A1*(A2+A3)**A6)/((A4+A5)**A7))
C     OR A POLYNOMIAL OF THE 5TH DEGREE, IF NDFQ = 1
C     FNEW = A2+A3*A1+A4*A1**2+A5*A1**3+A6*A1**4+A7*A1**5
      DIMENSION A(7)
      FNEW = -0.0
      IF(NPOLY .NE. 0) GO TO 50
      A(2) = A(2) + A(3)
      IF( A(2) .LE. 0. ) GO TO 10
      A(2) = A(2)**A(6)
      GO TO 20
10  N6FIX = IFIX( A(6) )
      IF( A(6) .NE. FLOAT(N6FIX) ) RETURN
      A(2) = A(2)**N6FIX
20  A(4) = A(4) + A(5)
      IF( A(4) .EQ. 0. ) RETURN
      IF( A(4) .LT. 0. ) GO TO 30
      A(4) = A(4)**A(7)
      GO TO 40
30  N7FIX = IFIX( A(7) )
      IF( A(7) .NE. FLOAT(N7FIX) ) RETURN
      A(4) = A(4)**N7FIX
40  FNEW = A(1) * A(2) / A(4)
      RETURN
50  FNEW = A(2) + A(1)*( A(3) + A(1)*( A(4) + A(1)*
1     ( A(5) + A(1)*( A(6) + A(1)*A(7) )))
      RETURN
      END

```

1441*

LEVEL 2 FEB 67

CS/360 FORTRAN H

DATE 67.230/04.16.30

COMPILER OPTIONS - NAME= MAIN,CPT=02,LINCCNT=72,SOURCE,BCD,NOLIST,DECK,LOAD,MAP,NOEDIT,IO

```

ISN 0002      SUBROUTINE WNAME(A,NV)
C
C      READ INPUT FOR PLOTTING THE PLCT DATA TAPE FOR W74740
C
C      A(I) = CHANNEL NAME FOR I-TH DATA CHANNEL VALUE OF ARRAY Y, WHERE
C              Y IS AN ARGUMENT OF WDISK
C      NV = NUMBER OF CHANNELS SUPPLIED AT EACH TIME BY CALLING PROGRAM
C           (MAX OF 1100)
C      INDA(K,N) = FOR TIME INTERVAL N, THE INDEX IN THE A ARRAY OF THE
C                   K-TH UNIQUE CHANNEL REQUESTED FOR PLOTTING (IN THE
C                   ORDER OF REQUEST AND THUS MULTIPLYOT INPUT)
C
C      NTIME = NUMBER OF TIME INTERVALS (MAX OF 4)
C      TMIN(N) = MINIMUM TIME OF INTERVAL N
C                 (.1 SEC)
C      TMAX(N) = MAXIMUM TIME OF INTERVAL N
C      NMPLT(N) = NUMBER OF MULTIPLYOTS FOR TIME INTERVAL N (MAX OF 500)
C      JYSC = NUMBER OF VERTICAL SCALES FOR MULTIPLYOT M OF TIME
C                INTERVAL N (MAX OF 3)
C      YMIN = MINIMUM VERTICAL SCALE VALUE FOR SCALE J OF
C                MULTIPLYOT M
C      YMAX = MAXIMUM VERTICAL SCALE VALUE
C      KGROUP = NUMBER OF GROUPS USING SCALE J OF MULTIPLYOT M OF TIME
C                INTERVAL N
C                (MAX OF 10 --- SEE NOTE FOR FURTHER RESTRICTIONS)
C      NCHAN = NUMBER OF CHANNELS (SEE NOTE ON RESTRICTIONS)
C      CHAN(IKJ) = CHANNEL NAME I FOR GROUP K, SCALE J, MULTIPLYOT M,
C                   TIME INTERVAL N. THE CHANNELS FOR EACH MULTIPLYOT
C                   ARE PACKED CONSECUTIVELY IN STORAGE.
C      *NOTE* - SUM OF NCHAN VALUES FOR A SINGLE MULTIPLYOT (ALL GROUPS
C                AND VERTICAL SCALES) HAS A MAX. OF 32
C      *NOTE* - FOR ANY VERTICAL SCALE, SUM OF NCHAN VALUES AND S*KGROUP
C                HAS A MAX. OF 62
C      NUM = NUMBER OF TIME VALUES FOR TIME INTERVAL (MAX OF 500)
C      DT(N) = TIME STEP FOR TIME INTERVAL N (.1 SEC)
C      NCTOT(N) = NUMBER OF CHANNELS USED FOR PLOTS FOR TIME INTERVAL N
C
ISN 0003      DIMENSION TMIN(4),TMAX(4), DT(4), NCTOT(4)
ISN 0004      DIMENSION TNEXT(4),ICRD(4)
ISN 0005      DIMENSION ICCUNT(4)
ISN 0006      INTEGER*2 INDA(1100,4)
ISN 0007      REAL*8 A(1100), CHAN(32)
C
C      READ INPUT FOR EACH TIME INTERVAL
C      SAVE INFORMATION NEEDED FOR DISK STORAGE ONLY
C
ISN 0008      READ (5,1000) NTIME
ISN 0009      IF (NTIME.GT.4) GO TO 255
ISN 0011      DO 250 N=1,NTIME
C
C      INPUT MIN TIME AND MAX TIME FOR TIME INTERVAL
C
ISN 0012      READ (5,1050) I1,I2
ISN 0013      TMIN(N)=I1
ISN 0014      TMAX(N)=I2
ISN 0015      IF (TMIN(N).GT.TMAX(N)) GO TO 260
C
C      DETERMINE TIME STEP USED FOR TIME INTERVAL N
C
ISN 0017      L=1
ISN 0018      DTMM=TMAX(N)-TMIN(N)
ISN 0019      10 NUM=DTMM/FLOAT(L)+.01
ISN 0020      IF (NUM.LE.499) GO TO 20
ISN 0022      L=L+1
ISN 0023      GO TO 10
ISN 0024      20 DT(N)=FLOAT(L)
ISN 0025      NUM=NUM+1
C
C      READ MULTIPLYOT NUMBER FOR TIME INTERVAL
C
ISN 0026      READ (5,1000) MPLT

```

PROGRAM FOR READING INPUT FOR PLOTTING THE PLOT DATA TAPE

30 of 87
PAGE 002

```

ISN 0027      IF( MPlot.GT.500)GO TO 265
C
C      READ CARDS 5-10 FOR EACH MULTIFLCT
C
ISN 0029      NCT=0
ISN 0030      DO 225 M=1,MPlot
ISN 0031      NC=0
ISN 0032      READ (5,1030)DUM
ISN 0033      READ (5,1000)JYSCL
ISN 0034      IF(JYSCL.GT.3)GO TO 270
C
C      READ CARDS 7-10 FOR EACH VERTICAL SCALE
C
ISN 0036      DO 110 J=1,JYSCL
ISN 0037      READ (5,1010)DUM,DUM,KGROUP
ISN 0038      IF(KGROUP.GT.10)GO TO 275
C
C      READ CARDS 8-10 FOR EACH GROUP
C
ISN 0040      NLINE=5*KGROUP
ISN 0041      DO 100 K=1,KGROUP
ISN 0042      READ (5,1000)NCHAN
ISN 0043      NCKM=NC+1
ISN 0044      NC=NC+NCHAN
ISN 0045      IF(NC.GT.32)GO TO 280
ISN 0047      NLINE=NLINE+NCHAN
ISN 0048      IF(NLINE.GT.62)GO TO 285
ISN 0050      READ (5,1020)(CHAN(I),I=NCKM,NC)
ISN 0051      READ (5,1030)DUM
ISN 0052      100 CONTINUE
ISN 0053      110 CONTINUE
C
C      GENERATE ARRAY OF CHANNEL INDICES FROM INPUT FOR EACH PLOT
C
C      FIND INPUT CHANNEL NAME IN ARRAY A
C
ISN 0054      DO 200 I=1,NC
ISN 0055      DO 120 J=1,NV
ISN 0056      IF(CHAN(I).EQ.A(J))GO TO 140
ISN 0058      120 CONTINUE
ISN 0059      GO TO 290
C
C      TEST IF CHANNEL PREVIOUSLY REQUESTED
C
ISN 0060      140 IF(NCT.EQ.0)GO TO 180
ISN 0062      DO 160 K=1,NCT
ISN 0063      IF(J.EQ.INDA(K,N))GO TO 200
ISN 0065      160 CONTINUE
ISN 0066      180 NCT=NCT+1
ISN 0067      INDA(NCT,N)=J
ISN 0068      200 CONTINUE
ISN 0069      225 CONTINUE
ISN 0070      NCTOT(N)=NCT
C
C      INITIAL WRITE ON DISK FOR TIME INTERVAL N
C      RECORD 1 - NUMBER OF CHANNELS, NUMBER OF TIME VALUES
C      RECORD 2,3 - CHANNEL NAMES A(J)
C
ISN 0071      NW=N
ISN 0072      WRITE (NW)NCTOT(N),NLM
ISN 0073      IF(NCT.GT.904)GO TO 235
ISN 0075      WRITE (NW) (A(INDA(NC,N)),NC=1,NCT)
ISN 0076      GO TO 250
ISN 0077      235 WRITE (NW) (A(INDA(NC,N)),NC=1,904)
ISN 0078      WRITE (NW)(A(INDA(NC,N)),NC=905,NCT)
ISN 0079      250 CONTINUE
ISN 0080      GO TO 320
C
C      ERROR MESSAGES
C
ISN 0081      255 WRITE (6,2980)NTIME
ISN 0082      STOP

```

PROGRAM FOR READING INPUT FOR PLOTTING THE PLOT DATA TAPE

31 of 87
PAGE 003

```

ISN 0083      260 WRITE (6,2990)N,TMIN(N),TMAX(N)
ISN 0084      STOP
ISN 0085      265 WRITE (6,3000)N,MPL0T
ISN 0086      STOP
ISN 0087      270 WRITE (6,3010)N,M,JYSCL
ISN 0088      STOP
ISN 0089      275 WRITE (6,3020)N,M,J,KGROUP
ISN 0090      STOP
ISN 0091      280 WRITE (6,3030)N,M,J,K,NC
ISN 0092      STOP
ISN 0093      285 WRITE (6,3040)N,M,J,K,NLINE
ISN 0094      STOP
ISN 0095      290 WRITE (6,3050)N,M,CHAN(I)
ISN 0096      STOP

C
C      FORMAT STATEMENTS
C
ISN 0097      1000 FORMAT(12I6)
ISN 0098      1010 FORMAT(2E12.5,I6)
ISN 0099      1020 FORMAT(6(A8,4X))
ISN 0100      1030 FORMAT(18A4)
ISN 0101      1050 FORMAT(6I12)
ISN 0102      2980 FORMAT(1H0/65H THE MAXIMUM NUMBER OF TIME INTERVALS IS 4.  THE IN
ISN 0103      2990 FORMAT(1H0/18H FOR TIME INTERVAL ,I3,1X,51H THE MINIMUM TIME IS GR
ISN 0104      3000 FORMAT(1H0/18H FOR TIME INTERVAL ,I3,1X,61H THE MAXIMUM NUMBER OF M
ISN 0105      3010 FORMAT(1H0/63H THE MAXIMUM NUMBER OF VERTICAL SCALES FOR EACH MULT
ISN 0106      3020 FORMAT(1H0/60H THE MAXIMUM NUMBER OF GROUPS FOR EACH VERTICAL SCAL
ISN 0107      3030 FORMAT(1H0/57H THE MAXIMUM NUMBER OF CHANNELS FOR EACH MULTIPL0T I
ISN 0108      3040 FORMAT(1H0/79H THE MAXIMUM VALUE OF THE SUM OF (NO. OF CHANNELS)
ISN 0109      3050 FORMAT(1H0/18H FOR TIME INTERVAL ,I2,1X,10H, MULTIPL0T ,I4,1X,13H,T
C
C      INITIALIZE VARIABLES FOR WDISK
C
C      TNEXT(N) = NEXT TIME USED FOR PLCTS OF TIME INTERVAL N
C      IORD(N)= INDICES FOR TIME INTERVALS ORDERED BY ASCENDING TMIN(N)
C      NORD = NUMBER OF VALUES CURRENTLY IN IORD ARRAY
C      (NUMBER OF TIME INTERVALS TO COMPLETE DATA STORAGE FOR)
C      NN = CURRENT TIME INTERVAL BEING CONSIDERED
ISN 0110      320 CONTINUE
ISN 0111      DO 330 N=1,NTIME
ISN 0112      TNEXT(N)=TMIN(N)
ISN 0113      IORD(N)=N
ISN 0114      ICOUNT(N)=0
ISN 0115      330 CONTINUE
ISN 0116      NTIMM=NTIME-1
ISN 0117      IF(NTIMM.EQ.0) GO TO 395
ISN 0118      DO 390 N=1,NTIMM
ISN 0119      I=IORD(N)
ISN 0120      NP=N+1
ISN 0121      DO 360 N2=NP,NTIME
ISN 0122      II=IORD(N2)
ISN 0123      IF(TMIN(I).LE.TMIN(II))GO TO 360
ISN 0124      I=II
ISN 0125      360 CONTINUE
ISN 0126      IORD(N)=I
ISN 0127
ISN 0128

```

```

ISN 0129      IORD(I)=N
ISN 0130      390 CONTINUE
ISN 0131      395 CONTINUE
ISN 0132      NORD=NTIME
ISN 0133      RETURN
C
ISN 0134      ENTRY WDISK(itime,y)
ISN 0135      TIME=itime
C
C      TEST TO DETERMINE INTERVALS USING CURRENT TIME
C      STORE PLOT DATA FOR TIME INTERVALS USING CURRENT TIME
C      WRITE RECORD ON DISK FOR TIME INTERVAL NN
C
C      TIME = CURRENT TAPE TIME
C      Y(I) = CURRENT CHANNEL DATA
C      INORD = INDEX IN ORDERED TIME INTERVAL ARRAY FOR CURRENT INTERVAL
C
ISN 0136      DIMENSION Y(1100)
C
C      TEST IF DISK DATA SETS COMPLETE FOR EACH TIME INTERVAL
C
ISN 0137      IF(NORD.EQ.0)RETURN
ISN 0139      INORD=1
ISN 0140      400 NN=IORD(INORD)
ISN 0141      IF(TIME.LT.TMIN(NN))RETURN
ISN 0143      IF(TIME.LT.TNEXT(NN))GO TO 600
C
C      WRITE SELECTED DATA ON DISK (CORRESPONDING TO TIME INTERVAL) WITH
C      ONE RECORD OF PARAMETER VALUES FOR A GIVEN TIME FORMING A RECORD
C
ISN 0145      NC=NCTOT(NN)
ISN 0146      NW=NN
ISN 0147      WRITE (NW)TIME,( Y(INDA(I,NN)),I=1,NC)
ISN 0148      460 CONTINUE
C
C      CHECK IF PLOTS FOR TIME INTERVAL COMPLETED
C
ISN 0149      IF(TIME.GE.TMAX(NN))GO TO 700
C
C      INCREMENT NEXT-TIME ARRAY
C
ISN 0151      500 CONTINUE
ISN 0152      TNEXT(NN)=TNEXT(NN)+DT(NN)
ISN 0153      IF(TIME.LT.TNEXT(NN))GO TO 600
ISN 0155      ICOUNT(NN)=1
ISN 0156      TNEX=TNEXT(NN)-DT(NN)
ISN 0157      WRITE (6,6020)TNEX,TMIN(NN),TMAX(NN)
ISN 0158      6020 FORMAT(1H0/ 9H0THE TIME ,E11.4 ,1X,43H(.1 SEC) WAS NOT INCLUDED 0
        IN THE DATA TAPE. /51H THIS TIME HAS BEEN DELETED FROM TIME INTERVA
        2L FROM ,E11.4 ,2HTO ,E11.4)
        GO TO 500
ISN 0159
C
C      IF ADDITIONAL TIME INTERVALS EXIST, CHECK NEXT INTERVAL
C      IF NOT, RETURN TO CALLING PROGRAM
C
ISN 0160      600 CONTINUE
ISN 0161      IF(NORD.EQ.INORD)RETURN
ISN 0163      INORD=INORD+1
ISN 0164      GO TO 400
C
C      PLOT DATA STORAGE COMPLETE FOR TIME INTERVAL
C      ELIMINATE TIME INTERVAL FROM ORDERED INTERVAL ARRAY AND CONTINUE
C
ISN 0165      700 CONTINUE
ISN 0166      NORD=NORD-1
ISN 0167      IF(NORD.LT.INORD)RETURN
ISN 0169      DO 800 I=INORD,NORD
ISN 0170      IORD(I)=IORD(I+1)
ISN 0171      800 CONTINUE
ISN 0172      GOTO 400
C
ISN 0173      ENTRY WMAX(itime)

```

```

ISN 0174      TIME=TIME
C
C      TEST IF ALL DATA SETS COMPLETE
C      IF NOT, WRITE LIST OF INCOMPLETE TIME INTERVALS
ISN 0175      IF (NORD.NE.0) WRITE (6,0000) TIME
ISN 0177      6000 FORMAT(1H0/ 43H0THE MAXIMUM TIME SUPPLIED FOR PLOTTING WAS, F0.0,
              11X, 8H(.1 SEC) /68H0THE FOLLOWING TIME INTERVALS REQUESTED FOR PLO
              2TTING WERE INCOMPLETE / 1H0,2X,4HTIME ,11X,4HTIME , 17X,4HTIME /
              39H INTERVAL ,7X,7HMINIMUM ,14X,7HMAXIMUM /7H INDEX ,8X,8H(.1 SEC)
              4 ,5X,9HREQUESTED ,8X,6HACTUAL )
ISN 0178      DO 900 NN=1,NTIME
ISN 0179      IF (NORD.E).0) GO TO 820
ISN 0181      DO 810 K=1,NORD
ISN 0182      IF (IORD(K).NE.NN) GO TO 810
ISN 0184      TNEXT(NN)=TNEXT(NN)-DT(NN)
ISN 0185      WRITE (6,0010) NN,TMIN(NN),TMAX(NN),TNEXT(NN)
ISN 0186      6010 FORMAT(1H ,14.4X,3F14.0)
ISN 0187      GO TO 830
ISN 0188      810 CONTINUE
ISN 0189      820 IF (ICOUNT(NN).EQ.0) GO TO 900
ISN 0191      830 CLNTINUE
C
C      WRITE DUMMY RECORD WITH TIME=-TIME FOR INCOMPLETE TIME INTERVALS
C
ISN 0192      NW=NN
ISN 0193      TIME=-TIME
ISN 0194      NC=NC+TOT(NN)
ISN 0195      WRITE (NW) TIME,(Y(J),J=1,NC)
ISN 0196      900 CONTINUE
ISN 0197      RETURN
ISN 0198      END

```

PROGRAM FOR READING INPUT FOR PLOTTING THE PLOT DATA TAPE

34 of 87

PAGE 006

NAME	TAG	TYPE	ADD.	NAME	TAG	TYPE	ADD.	WNAME /	NAME	TAG	TYPE	ADD.	NAME	TAG	TYPE	ADD.
A	XR	R*8	000000	I	I*4	000040		J	I*4	000044		K	I*4	000048		
L		I*4	00004C	M	I*4	000050		N	I*4	000054		Y	XR	R*4	000000	
DT		R*4	0000CC	II	I*4	000058		II	I*4	00005C		I2	I*4	000060		
NC		I*4	000064	NN	I*4	000068		NP	I*4	00006C		NV	I*4	000070		
NW		I*4	000074	N2	I*4	000078		DUM	R*4	00007C		NCT	I*4	000080		
NUM		I*4	000084	TIM	R*4	000088		CHAN	R*8	000140		DTMM	R*4	00008C		
INDA		I*2	000240	IORD	I*4	0000DC		NCKM	I*4	000090		NGRD	I*4	000094		
TIME		R*4	000098	TMAX	R*4	0000EC		TMIN	R*4	0000FC		TNEXT	R*4	00009C		
WMAX		R*4	0000A0	*DISK	R*4	0000A4		WNAME	R*4	N.R.		TNEXT	R*4	00010C		
NCHAN		I*4	0000A8	NC TOT	I*4	00011C		NLINE	I*4	0000AC		NTIME	I*4	0000B0		
NTIMM		I*4	0000B4	MPLGT	I*4	0000B8		JYSCl	I*4	0000BC		INORD	I*4	0000C0		
ITIME		I*4	0000C4	IBCDM# XF	I*4	000000		KGROUP	I*4	0000C8		ICOUNT	I*4	00012C		

LABEL	ADDR	LABEL	ADDR
100000	002C6C	100001	002CB6
100004	002D8A	100010	002DAE
100006	002E6C	100007	002E96
100010	002F28	100011	002F44
100014	002FC0	100015	002FCC
100017	003014	100018	00304C
100020	00308A	100019	00308E
160	0030BE	180	0030C2
100025	003112	100026	003158
235	0031C0	100029	003202
100032	00327C	250	003274
200	0032D2	265	003316
280	0033CA	285	003416
100034	0034CE	330	0034DE
100037	00350C	100038	003532
100040	003558	390	00356C
100043	0035FE	100044	003634
100046	003686	100047	00368E
100050	0037C4	500	003714
100052	00377C	100053	003784
100055	0037C0	100056	0037D6
100058	0037E6	100059	003848
100062	0038D2	100063	0038DC
810	003938	820	00393C
100068	003980	900	003984

***** END OF COMPILEION *****

PROGRAM FOR READING INPUT FOR PLOTTING THE PLOT DATA TAPE

LABEL	ADDR	LABEL	ADDR
100002	002CE0	100003	002CF4
100005	002E02	20	002E08
100008	002EE4	100009	002EF0
100012	002F80	100013	002F9C
100	002FEC	110	003002
100019	003070	120	003040
100021	003098	100022	0030B6
200	0030E4	225	0030FC
100027	0031A4	100028	0031D8
100030	003218	100031	00325C
100033	0032A4	255	0032A8
270	00334A	275	003386
290	003462	320	0034A6
100035	0034E4	100036	0034F4
100039	00354E	360	003550
395	003576	100042	003586
100045	00363C	400	003668
100048	00369A	100049	0036EE
100051	003732	600	003774
700	003798	100054	0037D8
800	0037DE	100057	0037E2
100060	00387E	100061	00389C
100064	0038E6	100065	0039F0
830	003946	100067	003974
100069	0039A2		

LEVEL 2 FEB 67

OS/360 FORTRAN H

DATE 67.236/12.39.33

COMPILER OPTIONS - NAME= MAIN,OPT=00,LINECNT=72,SOURCE,BCD,NOLIST,NODECK,LOAD,MAP,NCEDIT,IO

```

ISN 0002      BLOCK DATA
C*****
ISN 0003      REAL*8 DNAM,CDNAM
ISN 0004      COMMON/BLK1/DNAM(1100),CDNAM(17,200),COMENT(17,200,10)
ISN 0005      COMMON/BLK2/ NMAX(17), PDAT(5,200), LDAT(17,200),
1             DDAT(5,1100), LDSK(1100), KUNC, KCRD
ISN 0006      COMMON/BLK6/CGYTTL(18,17)
ISN 0007      DIMENSION
1             CGY1 (18), CGY2 (18), CGY3 (18), CGY4 (18), CGY5 (18),
1             CGY6 (18), CGY7 (18), CGY8 (18), CGY9 (18), CGY10(18),
1             CGY11(18), CGY12(18), CGY13(18), CGY14(18), CGY15(18),
1             CGY16(18), CGY17(18)
ISN 0008      EQUIVALENCE
1             (CGYTTL(1, 1),CGY1 (1)), (CGYTTL(1, 2),CGY2 (1)),
1             (CGYTTL(1, 3),CGY3 (1)), (CGYTTL(1, 4),CGY4 (1)),
1             (CGYTTL(1, 5),CGY5 (1)), (CGYTTL(1, 6),CGY6 (1)),
1             (CGYTTL(1, 7),CGY7 (1)), (CGYTTL(1, 8),CGY8 (1)),
1             (CGYTTL(1, 9),CGY9 (1)), (CGYTTL(1,10),CGY10(1)),
1             (CGYTTL(1,11),CGY11(1)), (CGYTTL(1,12),CGY12(1)),
1             (CGYTTL(1,13),CGY13(1)), (CGYTTL(1,14),CGY14(1)),
1             (CGYTTL(1,15),CGY15(1)), (CGYTTL(1,16),CGY16(1)),
1             (CGYTTL(1,17),CGY17(1))
C*****
ISN 0009      DATA KUNC/0/, KCRD/0/
ISN 0010      DATA CDNAM/3400*
ISN 0011      DATA
1             CGY1      /*A.  L I Q U I D  H Y D R O G E N  S Y S T E M
2             * * *
3             CGY2      /*B.  P U M P  A N D  P U M P  D I S C H A R G E
4             S Y S T E M
5             CGY3      /*C.  N O Z Z L E  * * *
6
7             CGY4      /*D.  R E F L E C T O R  * * *
8
9             CGY5      /*E.  S H I E L D  A N D  D U M E  S H I E L D
1
1             CGY6      /*F.  T H R U S T  C H A M B E R
2
3             CGY7      /*G.  T U R B I N E  F E E D  S Y S T E M
4
5             CGY8      /*H.  T U R B I N E  A N D  T U R B I N E  E X H
6             A U S T
7             CGY9      /*I.  D U C T  * * *
8
ISN 0012      DATA
9             CGY10     /*J.  S T E A M  G E N E R A T O R S
1
1             CGY11     /*K.  E X T E R N A L  S H I E L D  A N D  T E S
2             T C E L L
3             CGY12     /*L.  L I Q U I D  H Y D R O G E N  C O O L D O W
4             N S Y S T E M
5             CGY13     /*M.  G A S E O U S  H Y D R O G E N  S Y S T E M
6
7             CGY14     /*N.  G A S E O U S  H E L I U M  S Y S T E M
8
9             CGY15     /*O.  C O N T R O L  S Y S T E M S
1
1             CGY16     /*P.  M I S C E L L A N E O U S
2
3             CGY17     /*Q.  C O R E  * * * * *
4
ISN 0013      END

```


PROGRAM FOR PRINTING OUT STEADY-STATE SUMMARIES

37 of 87

PAGE 002

NAME	TAG	TYPE	ADD.	NAME	TAG	TYPE	ADD.	MAIN	NAME	TAG	TYPE	ADD.	NAME	TAG	TYPE	ADD.
CGY1	CE	R*4	N.R.	CGY2	CE	R*4	N.R.		CGY3	CE	R*4	N.R.	CGY4	CE	R*4	N.R.
CGY5	CE	R*4	N.R.	CGY6	CE	R*4	N.R.		CGY7	CE	R*4	N.R.	CGY8	CE	R*4	N.R.
CGY9	CE	R*4	N.R.	DDAT	C	R*4	N.R.		DNAM	C	R*4	N.R.	KCRD	C	I*4	N.R.
KUNC	C	I*4	N.R.	LDAT	C	I*4	N.R.		LDSK	C	I*4	N.R.	NMAX	C	I*4	N.R.
PDAT	C	R*4	N.R.	CDNAM	C	R*4	N.R.		CGY10	CE	R*4	N.R.	CGY11	CE	R*4	N.R.
CGY12	CE	R*4	N.R.	CGY13	CE	R*4	N.R.		CGY14	CE	R*4	N.R.	CGY15	CE	R*4	N.R.
CGY16	CE	R*4	N.R.	CGY17	CE	R*4	N.R.		IBCOM#	XF		000000	COMENT	C	R*4	N.R.
CGYTTL	CE	R*4	N.R.													

***** END OF COMPILATION *****

PROGRAM FOR PRINTING OUT STEADY-STATE SUMMARIES

38 of 87

LEVEL 2 FEB 67

OS/360 FORTRAN H

DATE 67.236/12.40.00

COMPILER OPTIONS - NAME= MAIN,OPT=00,LINECNT=72,SOURCE,BCD,NOLIST,NODECK,LJAD,MAP,NGEDIT,IO

```

C*****
C  MAIN PROGRAM
C*****
      REAL*8 DNAM,CDNAM
      COMMON/BLK1/DNAM(1100),CDNAM(17,200),COMFNT(17,200,10)
      COMMON/BLK2/ NMAX(17), PDAT(5,200), LDAT(17,200),
1      DDAT(5,1100), LDSK(1100), KUNC, KCRD
      COMMON/BLK3/ TST(5), TFIN(5), NTST(5), NTFIN(5), ITST, ITFIN
      COMMON/BLK4/ NCAT(17), NLIN, NWCAT, NDNAM, NSET, NDUAP
      COMMON/BLK5/ DATE(2)
      COMMON/BLK6/CGYTTL(18,17)
      DIMENSION
1      CGY1 (18), CGY2 (18), CGY3 (18), CGY4 (18), CGY5 (18),
1      CGY6 (18), CGY7 (18), CGY8 (18), CGY9 (18), CGY10(18),
1      CGY11(18), CGY12(18), CGY13(18), CGY14(18), CGY15(18),
1      CGY16(18), CGY17(18)
      EQUIVALENCE
1      (CGYTTL(1, 1),CGY1 (1)), (CGYTTL(1, 2),CGY2 (1)),
1      (CGYTTL(1, 3),CGY3 (1)), (CGYTTL(1, 4),CGY4 (1)),
1      (CGYTTL(1, 5),CGY5 (1)), (CGYTTL(1, 6),CGY6 (1)),
1      (CGYTTL(1, 7),CGY7 (1)), (CGYTTL(1, 8),CGY8 (1)),
1      (CGYTTL(1, 9),CGY9 (1)), (CGYTTL(1,10),CGY10(1)),
1      (CGYTTL(1,11),CGY11(1)), (CGYTTL(1,12),CGY12(1)),
1      (CGYTTL(1,13),CGY13(1)), (CGYTTL(1,14),CGY14(1)),
1      (CGYTTL(1,15),CGY15(1)), (CGYTTL(1,16),CGY16(1)),
1      (CGYTTL(1,17),CGY17(1))
C*****
      CALL RDCARD(1)
      CALL RDISK(1,0)
10     CALL RDCARD(2)
      NWCAT = 1
      NLIN = 0
      DO 30 I=1,NSET
      CALL RDISK(2,I)
      DO 40 I=1,17
      IF(NCAT(I) .EQ. 0) GO TO 50
      CALL DXFER(NCAT(I),NSET)
      CALL PRDAT(NCAT(I))
40     NWCAT = 1
      DO 50 I=1,NSET
      CALL ERROR(6)
      GO TO 10
      END
ISN 0002
ISN 0003
ISN 0004
ISN 0005
ISN 0006
ISN 0007
ISN 0008
ISN 0009
ISN 0010
ISN 0011
ISN 0012
ISN 0013
ISN 0014
ISN 0015
ISN 0016
ISN 0017
ISN 0018
ISN 0019
ISN 0021
ISN 0022
ISN 0023
ISN 0024
ISN 0025
ISN 0026

```

PROGRAM FOR PRINTING OUT STEADY-STATE SUMMARIES

39 of 87

PAGE 002

NAME	TAG	TYPE	ADD.	NAME	TAG	TYPE	ADD.	MAIN /	NAME	TAG	TYPE	ADD.	NAME	TAG	TYPE	ADD.
I		I*4	000028	TST	C	R*4	N.R.		CGY1	CE	R*4	N.R.	CGY2	CE	R*4	N.R.
CGY3	CE	R*4	N.R.	CGY4	CE	R*4	N.R.		CGY5	CE	R*4	N.R.	CGY6	CE	R*4	N.R.
CGY7	CE	R*4	N.R.	CGY3	CE	R*4	N.R.		CGY9	CE	R*4	N.R.	DATE	C	R*4	N.P.
DDAT	C	R*4	N.R.	DNAM	C	R*8	N.R.		ITST	C	I*4	N.R.	KCRD	C	I*4	N.P.
KUNC	C	I*4	N.R.	LDAT	C	I*4	N.R.		LDISK	C	I*4	N.R.	NCAT	C	I*4	000000
NLIN	C	I*4	000044	NMAX	C	I*4	N.R.		NSET	C	I*4	000050	NTST	C	I*4	N.R.
PDAT	C	R*4	N.R.	TFIN	C	R*4	N.R.		RDISK	XF	R*4	000000	NDNAM	C	I*4	N.R.
NDUMP	C	I*4	N.R.	NTFIN	C	I*4	N.R.		NWCAT	C	I*4	000048	ITFIN	C	I*4	N.R.
ERROR	XF	R*4	000000	DXFER	XF	R*4	000000		CDNAM	C	R*8	N.R.	CGY10	CE	R*4	N.R.
CGY11	CE	R*4	N.R.	CGY12	CE	R*4	N.R.		CGY13	CE	R*4	N.R.	CGY14	CE	R*4	N.R.
CGY15	CE	R*4	N.R.	CGY15	CE	R*4	N.R.		CGY17	CE	R*4	N.R.	RDCARD	XF	R*4	000000
PRTDAT	XF	R*4	000000	IBCOM#	XF	I*4	000000		COMENT	C	R*4	N.R.	CGYTTL	CE	R*4	N.R.

PROGRAM FOR PRINTING OUT STEADY-STATE SUMMARIES

40 of 87

PAGE 003

LABEL	ADDR	LABEL	ADDR
100000	00016C	10	000198
100002	0001E2	100003	0001FA

LABEL	ADDR
30	0001B2
40	00023E

LABEL	ADDR
100001	00010A
50	00025C

***** END OF COMPILATION *****

LEVEL 2 FEB 67

OS/360 FORTRAN H

DATE 67.236/12.40.06

COMPILER OPTIONS - NAME= MAIN,OPT=00,LINECNT=72,SOURCE,BCD,NOLIST,NODECK,LOAD,MAP,NOEDIT,ID

```

      SUBROUTINE RDISK(NOP,NCOL)
      REAL*8 DNAM,CDNAM
      COMMON/BLK1/DNAM(1100),CDNAM(17,200),COMMENT(17,200,10)
      COMMON/BLK2/ NMAX(17), PDATE(5,200), LDAT(17,200),
1      DDATE(5,1100), LDSK(1100), KUNC, KCRD
      COMMON/BLK3/ TST(5), TFIN(5), NTST(5), NTFIN(5), ITST, ITFIN
      COMMON/BLK4/ NCAT(17), NLIN, NWCAT, NDNAM, NSET, NDUMP
      DIMENSION NPOINT(23)
      IF(NOP .EQ. 2) GO TO 40
C*****
C      READ OFF DISK THE ARRAY NPOINT WHICH INDICATES HOW MANY
C      ITEMS ARE CONTAINED ON THE FOLLOWING RECORDS.
C*****
      WRITE(6,5070)
10      READ(11,END=60)      KYWRD,NPOINT
      WRITE(6,5040) KYWRD,NPOINT
      IF(KYWRD .NE. 1) CALL ERROR(1)
      NLNAM = NPOINT(23)
C*****
C      READ THE DISK CHANNEL NAMES.
C*****
      WRITE(6,5080)
20      READ(11,END=70)      KYWRD, (DNAM(I),I=1,NDNAM)
      WRITE(6,5040) KYWRD
      WRITE(6,5050) (DNAM(I), I=1,NDNAM)
      IF(KYWRD .NE. 2) CALL ERROR(1)
      NLLOC = NPOINT(23)
C*****
C      READ THE LOCATION NUMBERS FOR THE DISK DATA.
C*****
      WRITE(6,5090)
30      READ(11,END=80)      KYWRD, (LDSK(I),I=1,NLLOC)
      WRITE(6,5040) KYWRD, (LDSK(I), I=1,NLLOC)
      IF(KYWRD .NE. 3) CALL ERROR(1)
      CALL NAMCEK
      RETURN
C*****
C      READ A DATA RECORD AND CHECK FOR THE RIGHT TIME.
C*****
40      ND = NPOINT(23)
50      READ(11,END=90)      KYWRD,ITST,ITFIN,(DDAT(NCOL,I),I=1,ND)
      IF(NDUMP .EQ. C) GO TO 55
      WRITE(6,5040) KYWRD,ITST,ITFIN
      WRITE(6,5060) (DDAT(NCOL,I), I=1,ND)
      IF((KYWRD .NE. 0) .AND. (KYWRD .NE. 10)) CALL ERROR(2)
      IF(ITST .LT. NTST(NCOL)) GO TO 50
      IF(ITFIN .EQ. 0) CALL ERROR(3)
      IF((ITST .EQ. NTST(NCOL)) .AND. (ITFIN .EQ. NTFIN(NCOL))) RETURN
      GO TO 50
60      WRITE(6,5000)
      WRITE(6,5040) NPOINT
      STOP 1111
70      WRITE(6,5010)
      WRITE(6,5050) DNAM
      STOP 2222
80      WRITE(6,5020)
      WRITE(6,5040) LDSK
      STOP 3333
90      WRITE(6,5030)
      WRITE(6,5060) DDATE
      STOP 4444
5000      FORMAT('1'5X'EOF IN POINTER RECORD.')
```

```

5010      FORMAT('1'5X'EOF IN DNAM RECORD.')
```

```

5020      FORMAT('1'5X'EOF IN LOCATION RECORD.')
```

```

5030      FORMAT('1'5X'EOF IN DDATE RECORDS.')
```

```

5040      FORMAT(10X,'KEYWORD = 'I2/(16X,10110))
```

```

5050      FORMAT(20X,10A10)
```

```

5060      FORMAT(9X,10F12.2)
```

```

5070      FORMAT('1'4X'-----POINTER NUMBERS-----'/)
```

```

5080      FORMAT('1'50X'-----DISK CHANNEL NUMBERS-----'/)
```

```

5090      FORMAT('1'44X'-----DISK DATA LOCATION NUMBERS-----'/)
      END
```

PROGRAM FOR PRINTING OUT STEADY-STATE SUMMARIES

42 of 87

PAGE 003

NAME	TAG	TYPE	ADD.	NAME	TAG	TYPE	ADD.	RDISK /	NAME	TAG	TYPE	ADD.	NAME	TAG	TYPE	ADD.
I		I*4	000028	ND		I*4	00002C		NOP		I*4	000030	TST	C	R*4	N.R.
DDAT	C	R*4	004504	DNAM	C	R*8	000000		ITST	C	I*4	000050	KCRD	C	I*4	N.R.
KUNC	C	I*4	N.R.	LDAT	C	I*4	N.R.		LDSK	C	I*4	009AF4	NCAT	C	I*4	N.R.
NCOL		I*4	000034	NLIN	C	I*4	N.R.		NLOC		I*4	000034	NMAX	C	I*4	N.R.
NSET	C	I*4	N.R.	NTST	C	I*4	000028		PDAT	C	R*4	N.R.	TFIN	C	R*4	N.R.
RDISK		R*4	N.R.	NDNAM	C	I*4	00004C		NDUMP	C	I*4	000054	NTFIN	C	I*4	00003C
NWCAT	C	I*4	N.R.	KYWRD		I*4	00003C		ITFIN	C	I*4	000054	ERRDR	XF	R*4	000000
CDNAM	C	R*6	N.R.	IBCOM#	XF	I*4	000000		NPOINT		I*4	000040	NAMCEK	XF	I*4	000000
COMENT	C	R*4	N.R.													

PROGRAM FOR PRINTING OUT STEADY-STATE SUMMARIES

43 of 87

PAGE 004

LABEL	ADDR	LABEL	ADDR	LABEL	ADDR	LABEL	ADDR
1000C0	000000	100001	0003B2	100002	0003C0	100003	000432
100004	000440	100005	000480	100006	0004AF	100007	0004F8
100008	000516	100009	000528	100010	000536	100011	000574
100012	00059E	100013	0005C4	100014	0005EE	100015	000600
100016	00060E	40	000620	50	000628	100017	00065C
100018	00069A	100019	0006AE	100020	0006F8	100021	000736
55	00073A	100022	000774	100023	000782	100024	00079C
100025	0007AC	100026	0007BA	100027	000808	100028	000810
60	000816	70	00085A	80	00089E	90	0008E2

***** END OF COMPILATION *****

PROGRAM FOR PRINTING OUT STEADY-STATE SUMMARIES

44 of 87

LEVEL 2 FEB 67

OS/360 FORTRAN H

DATE 67.236/12.40.12

COMPILER OPTIONS - NAME= MAIN,OPT=00,LINECNT=72,SOURCE,BCD,NOLIST,NODECK,LOAD,MAP,NOEDIT,ID

```

ISN 0002      SUBROUTINE DXFER(NCTG,NSET)
C*****
C      TRANSFER DISK DATA ARRAY DDAT INTO PRINT DATA ARRAY PDAT.
C      NCTG= CATEGORY NUMBER.
C      NSET= NUMBER OF SETS OR COLUMNS IN THE NCTG CATEGORY.
C      NMAX(NCTG)= TOTAL CHANNELS IN THE CATEGORY.
C*****
ISN 0003      COMMON/BLK2/ NMAX(17), PDAT(5,200), LDAT(17,200),
1              DDAT(5,1100), LDSK(1100), KUNC, KCRD
C*****
ISN 0004      DO 20 N=1,NSET
ISN 0005      NINCAT = NMAX(NCTG)
ISN 0006      DO 20 M=1,NINCAT
ISN 0007      PDAT(N,M) = DDAT(N,LDAT(NCTG,M))
ISN 0008      IF(LDAT(NCTG,M) .LE. 0) PDAT(N,M) = -0.0
ISN 0010      20 CONTINUE
ISN 0011      RETURN
ISN 0012      END
    
```


PROGRAM FOR PRINTING OUT STEADY-STATE SUMMARIES

45 of 87

PAGE 002

NAME	TAG	TYPE	ADD.
M		I*4	000010
KUNC	C	I*4	N.R.
NMAX	C	I*4	000000
NINCAT		I*4	000020

NAME	TAG	TYPE	ADD.
N		I*4	000014
LDAT	C	I*4	000FE4
NSET		I*4	00001C

NAME	TAG	TYPE	ADD.
DDAT	C	R*4	004504
LDSK	C	I*4	N.R.
PDAT	C	R*4	000044

NAME	TAG	TYPE	ADD.
KCRD	C	I*4	N.R.
NCTG		I*4	000018
DXFER		R*4	N.R.

PROGRAM FOR PRINTING OUT STEADY-STATE SUMMARIES

46 of 87

PAGE 013

LABEL	ADDR	LABEL	ADDR	LABEL	ADDR	LABEL	ADDR
100000	000000	100001	00010E	100002	000116	100003	000132
100004	0001C0	20	0001E8	100005	0001FE	100006	000214

***** END OF COMPILATION *****

LEVEL 2 FEB 67

OS/360 FORTRAN H

DATE 67.236/12.40.17

COMPILER OPTIONS - NAME= MAIN,OPT=00,LINECNT=72,SOURCE,BCD,NOLIST,NODECK,LOAD,MAP,NODEDIT,IO

```

ISN 0002      SUBROUTINE PRDATT(NCTG)
C*****
C      PRINT THE DATA FOR ONE CATEGORY.
C      NCTG= CATEGORY NUMBER.
C      NINCAT= NUMBER OF CHANNELS IN THE NCTG CATEGORY.
C      NSET= NUMBER OF SETS IN THE NCTG CATEGORY.
C      NWCAT=1 IMPLIES A NEW CATEGORY AT MID-PAGE AND REQUIRES A PRINT OF
C      THE CATEGORY TITLE, CGYTTL.
C      NWCAT=0 SUPPRESSES PRINT OF CATEGORY TITLE.
C*****
ISN 0003      REAL*8 DNAM,CDNAM
ISN 0004      COMMON/BLK1/DNAM(1100),CDNAM(17,200),COMENT(17,200,10)
ISN 0005      COMMON/BLK2/ NMAX(17), PDAT(5,200), LDAT(17,200),
1             DDAT(5,1100), LDSK(1100), KUNC, KCRD
ISN 0006      COMMON/BLK3/ TST(5), TFIN(5), NTST(5), NTFIN(5), ITST, ITFIN
ISN 0007      COMMON/BLK4/ NCAT(17), NLIN, NWCAT, NDNAM, NSET, NDUMP
ISN 0008      COMMON/BLK5/ DATE(2)
ISN 0009      COMMON/BLK6/CGYTTL(18,17)
C*****
ISN 0010      DIMENSION
1             CGY1 (18), CGY2 (18), CGY3 (18), CGY4 (18), CGY5 (18),
1             CGY6 (18), CGY7 (18), CGY8 (18), CGY9 (18), CGY10(18),
1             CGY11(18), CGY12(18), CGY13(18), CGY14(18), CGY15(18),
1             CGY16(18), CGY17(18)
ISN 0011      EQUIVALENCE
1             (CGYTTL(1, 1),CGY1 (1)), (CGYTTL(1, 2),CGY2 (1)),
1             (CGYTTL(1, 3),CGY3 (1)), (CGYTTL(1, 4),CGY4 (1)),
1             (CGYTTL(1, 5),CGY5 (1)), (CGYTTL(1, 6),CGY6 (1)),
1             (CGYTTL(1, 7),CGY7 (1)), (CGYTTL(1, 8),CGY8 (1)),
1             (CGYTTL(1, 9),CGY9 (1)), (CGYTTL(1,10),CGY10(1)),
1             (CGYTTL(1,11),CGY11(1)), (CGYTTL(1,12),CGY12(1)),
1             (CGYTTL(1,13),CGY13(1)), (CGYTTL(1,14),CGY14(1)),
1             (CGYTTL(1,15),CGY15(1)), (CGYTTL(1,16),CGY16(1)),
1             (CGYTTL(1,17),CGY17(1))
ISN 0012      DATA TO, TO*
C*****
ISN 0013      900 FORMAT('1' // 30X ' N E R V A   S T E A D Y - S T A T E   D A T
1A            ' S U M M A R Y '
2            / 104X 'TEST DATE '          2A4          / 50X
3            'P R O G R A M   E - 5 7 1 0 5' 25X 'G.G. STRUCEL, DEPT. 7830' / )
ISN 0014      910 FORMAT('0', 20X, 18A4)
ISN 0015      920 FORMAT(/62X 'C O N T R O L   R O O M   T I M E   I N T E R V A L -
1            ' S E C O N D S' / 54X, 5F15.1)
ISN 0016      930 FORMAT(47X'CHANNEL' 8X,A4,4(11X,A4))
ISN 0017      940 FORMAT(11X,'CHANNEL NAME'14X,'UNITS
ISN 0018      945 FORMAT(' ')
ISN 0019      950 FORMAT(1X, I3, ' ', 10A4,1X, A8, F13.2, 4F15.2)
C*****
ISN 0020      NINCAT = NMAX(NCTG)
ISN 0021      DO 50 M=1,NINCAT
ISN 0022      IF(NLIN .NE. 0) GO TO 10
C*****
C      THE FOLLOWING SIX STATEMENTS WRITE THE PAGE TOP TITLE.
C*****
ISN 0024      WRITE(6,900) DATE
ISN 0025      WRITE(6,910) (CGYTTL(I,NCTG), I=1,18)
ISN 0026      WRITE(6,920) (TST(I), I=1,NSET)
ISN 0027      WRITE(6,930) (TO,I=1,NSET)
ISN 0028      WRITE(6,940) (TFIN(I), I=1,NSET)
ISN 0029      WRITE(6,945)
ISN 0030      NLIN = NLIN+14
ISN 0031      GO TO 20
ISN 0032      10 IF(NWCAT .EQ. 0) GO TO 20
ISN 0033      WRITE(6,910) (CGYTTL(I,NCTG), I=1,18)
ISN 0034      NLIN = NLIN +2
ISN 0035      20 IF(NINCAT .LE. 0) GO TO 30
ISN 0036      WRITE(6,950) M, (COMENT(NCTG,M,L), L=1,10),CDNAM(NCTG,M),
ISN 0038      1             (PDAT(I,M), I=1,NSET)
ISN 0039      GO TO 40
ISN 0040      30 WRITE(6,945)
ISN 0041      40 NLIN = NLIN +1

```

PROGRAM FOR PRINTING OUT STEADY-STATE SUMMARIES

48 of 87

PAGE 002

```

ISN 0042      NWCAT = 0
C*****
C      IF THE LINE COUNTER, NLIN EXCEEDS 79, RESET TO ZERO AND PRINT THE
C      PAGE TOP TITLE.
C*****
ISN 0043      IF(NLIN .GT. 79) NLIN = 0
ISN 0045      50 CONTINUE
ISN 0046      RETURN
ISN 0047      END

```

PROGRAM FOR PRINTING OUT STEADY-STATE SUMMARIES

49 of 87

PAGE 003

NAME	TAG	TYPE	ADD.	NAME	TAG	TYPE	ADD.	PRTDAT	NAME	TAG	TYPE	ADD.	NAME	TAG	TYPE	ADD.
I		I*	000030	L		I*	000034		M		I*	000038	TD		R*	00003C
TST	C	R*	000000	CGY1	CF	R*	N.R.		CGY2	CE	R*	N.R.	CGY3	CE	R*	N.R.
CGY4	CL	R*	N.R.	CGY5	CF	R*	N.R.		CGY6	CE	R*	N.R.	CGY7	CF	R*	N.R.
CGY8	CE	R*	N.R.	CGY9	CE	R*	N.R.		DATE	C	R*	000000	DDAT	C	R*	N.R.
DNAM	C	R*	N.R.	ITST	C	I*	N.R.		KCRD	C	I*	N.R.	KUNC	C	I*	N.R.
LDAT	C	I*	N.R.	LUSK	C	I*	N.R.		NCAT	C	I*	N.R.	NCTG		I*	000040
NLIN	C	I*	000044	NMAX	C	I*	000000		NSET	C	I*	000050	NTST	C	I*	N.R.
PDAT	C	R*	000044	TFIN	C	R*	000014		NDNAM	C	I*	N.R.	NDUMP	C	I*	N.R.
NTFIN	C	I*	N.R.	NWCAT	C	I*	000048		ITFIN	C	I*	N.R.	CDNAM	C	R*	002260
CGY10	CE	R*	N.R.	CGY11	CE	R*	N.R.		CGY12	CE	R*	N.R.	CGY13	CE	R*	N.R.
CGY14	CE	R*	N.R.	CGY15	CE	R*	N.R.		CGY16	CE	R*	N.R.	CGY17	CE	R*	N.R.
IBCOM#	XF	I*	000000	PRTDAT		R*	N.R.		NINCAT		I*	000044	COMENT	C	R*	006CA0
CGYTTL	CE	R*	000000													

PROGRAM FOR PRINTING OUT STEADY-STATE SUMMARIES

50 of 87

PAGE 004

LABEL	ADDR
100000	000000
100004	000364
100008	00040C
10	0004AA
20	000526
100018	0005D0
100020	000656

LABEL	ADDR
100001	0002FE
100005	0003A2
100009	00042E
100012	0004BA
100015	000532
100019	000612
50	000662

LABEL	ADDR
100002	00031A
100006	0003C0
100010	00044C
100013	0004D4
100016	000554
30	00061C
100021	000678

LABEL	ADDR
100003	00032A
100007	0003EE
100011	00047A
100014	000512
100017	0005A2
40	000630

***** END OF COMPILATION *****

LEVEL 2 FEB 67

OS/360 FORTRAN H

DATE 67.236/12.40.24

COMPILER OPTIONS - NAME= MAIN,OPT=00,LINECNT=72,SOURCE,BCD,NOLIST,NODECK,LOAD,MAP,NOEDIT,IO

```

ISN 0002      SUBROUTINE NAMCEK
C*****
C SEARCHES FOR THE REQUESTED CHANNEL NAMES WHICH WERE INPUT ON CARDS
C FROM AMONG THE LIST OF CHANNEL NAMES ON THE SUMMARY DISK.
C IOK=1 ALL REQUESTED CHANNELS FOUND.
C IOK=0 AT LEAST ONE CHANNEL NOT FOUND.
C NMAX(ICAT)= NUMBER OF CHANNELS IN THE ICAT CATEGORY.
C NDNAM= TOTAL NUMBER OF CHANNEL NAMES ON DISK.
C LDAT(ICAT,J)= THE LOCATION NUMBER ON DISK DATA OF THE J CHANNEL
C OF THE ICAT CATEGORY.
C*****
C*****
ISN 0003      REAL*8 DNAM,CDNAM
ISN 0004      COMMON/BLK1/DNAM(1100),CDNAM(17,200),COMENT(17,200,10)
ISN 0005      COMMON/BLK2/ NMAX(17), PDAT(5,200), LDAT(17,200),
ISN 0006      1 DDAT(5,1100), LDSK(1100), KUNC, KCRD
ISN 0007      COMMON/BLK4/ NCAT(17), NLIN, NWCAT, NDNAM, NSET, NDUMP
ISN 0008      900 FORMAT('1'27X'CHANNEL SEARCH CHECK --- THE FOLLOWING REQUESTED'/
ISN 0009      1 27X'CHANNELS COULD NOT BE FOUND ON DISK.'//)
ISN 0010      910 FORMAT(30X,A8,2I15)
ISN 0011      920 FORMAT(40X, 'ALL CHANNELS PRESENT.'/'1')
C*****
ISN 0010      IOK = 1
ISN 0011      WRITE(6,900)
ISN 0012      DO 10 ICAT = 1,17
ISN 0013      IF(NMAX(ICAT).EQ. 0) GO TO 10
ISN 0015      NINCAT = NMAX(ICAT)
ISN 0016      DO 10 J=1,NINCAT
ISN 0017      DO 20 K=1,NDNAM
ISN 0018      IF(CDNAM(ICAT,J) .NE. DNAM(K)) GO TO 20
ISN 0020      LDAT(ICAT,J) = LDSK(K) -3
ISN 0021      GO TO 10
ISN 0022      20 CONTINUE
C*****
C IF THE REQUIRED CHANNEL NAME CDNAM CANNOT BE FOUND AMONG THE
C DISK NAMES, PRINT THE CHANNEL WITH ITS CATEGORY NUMBER AND SUB-
C SCRIPT NUMBER.
C*****
ISN 0023      LDAT(ICAT,J) = 0
ISN 0024      IOK = 0
ISN 0025      WRITE(6,910) CDNAM(ICAT,J), ICAT,J
ISN 0026      10 CONTINUE
ISN 0027      IF(IOK .EQ. 0) RETURN
ISN 0029      WRITE(6,920)
ISN 0030      RETURN
ISN 0031      END

```

PROGRAM FOR PRINTING OUT STEADY-STATE SUMMARIES

52 of 87

PAGE 002

NAME	TAG	TYPE	ADD.	NAME	TAG	TYPE	ADD.	NAMCEK	NAME	TAG	TYPE	ADD.	NAME	TAG	TYPE	ADD.
J		I*4	000020	K		I*4	000024	/	IOK		I*4	000028	DDAT	C	R*4	N.R.
DNAM	C	R*8	00C000	ICAT		I*4	00002C	/	KCRD	C	I*4	N.R.	KUNC	C	I*4	N.R.
LDAT	C	I*4	000FE4	LDSK	C	I*4	009AF4	/	NCAT	C	I*4	N.R.	NLIN	C	I*4	N.R.
NMAX	C	I*4	00C000	NSET	C	I*4	N.R.	/	PDAT	C	R*4	N.R.	NDNAM	C	I*4	00004C
NDUMP	C	I*4	N.R.	NWCAT	C	I*4	N.R.	/	CDNAM	C	R*8	002260	IBCOM#	XF	I*4	000000
NINCAT		I*4	000030	NAMCEK		I*4	N.R.	/	COMENT	C	R*4	N.R.				

PROGRAM FOR PRINTING OUT STEADY-STATE SUMMARIES

53 of 87

LABEL	ADDR	LABEL	ADDR
100000	000000	100001	000186
100004	000210	100005	000218
100007	000284	10	00032C
100010	000364	100011	00036C

***** END OF COMPILATION *****

LABEL	ADDR
100002	0001DC
100006	000258
100008	000342

LABEL	ADDR
100003	0001F4
20	00029A
100009	000358

PAGE 003

PROGRAM FOR PRINTING OUT STEADY-STATE SUMMARIES

54 of 87

LEVEL 2 FEB 67

DS/360 FORTRAN H

DATE 67.236/12.40.29

COMPILER OPTIONS - NAME= MAIN,OPT=00,LINECNT=72,SOURCE,BCD,NOLIST,NODECK,LOAD,MAP,NOEDIT,ID

```

ISN 0002      SUBROUTINE ERROR(NERR)
ISN 0003      C*****
ISN 0004      COMMON/BLK3/ TST(5), TFIN(5), NTST(5), NTFIN(5), ITST, ITFIN
ISN 0005      COMMON/BLK4/ NCAT(17), NLIN, NWCAT, NDNAM, NSET, NDUMP
ISN 0006      C*****
ISN 0007      910 FORMAT('1',' KEY WORD ON DISK RECORD INDICATES RECCRD OUT OF SEQUE
ISN 0008      INCE - CALL EXIT.')
```

```

ISN 0009      920 FORMAT('1',' INCORRECT KEYWORD ON DISK DATA RECORD, CONTINUE TIME
ISN 0010      1SEARCH.')
```

```

ISN 0011      930 FORMAT('1',' END OF TIME INTERVAL ON DISK IS ZERO - CALL EXIT.')
```

```

ISN 0012      940 FORMAT('1',' CLASSIFIED AND UNCLASSIFIED INFORMATION MAY NOT BE PR
ISN 0013      10CESSED IN THE SAME CASE OR JOB.')
```

```

ISN 0014      2 ' CHECK INPUT SYSTEM NUMBERS AND MAKE ALL REQUESTS UNDER THE PR
ISN 0015      30PER SECURITY CLASSIFICATION.')
```

```

ISN 0016      4 ' SKIP THIS CASE.')
```

```

ISN 0017      950 FORMAT(10X,4I15)
```

```

ISN 0018      960 FORMAT('1',37X' * * * * * E N D   O F   J O B   * * * * *')
```

```

ISN 0019      970 FORMAT('1',37X' * * * * * E N D   O F   C A S E   * * * * *'/'1')
```

```

ISN 0020      C*****
ISN 0021      GO TO ( 10,20,30,40,50,60) , NERR
ISN 0022      10 WRITE(6,910)
ISN 0023      WRITE(6,970)
ISN 0024      WRITE(6,960)
ISN 0025      GO TO 70
ISN 0026      20 WRITE(6,920)
ISN 0027      WRITE(6,950) ITST,ITFIN, (NTST(N), NTFIN(N), N=1,5)
ISN 0028      RETURN
ISN 0029      30 WRITE(6,930)
ISN 0030      WRITE(6,970)
ISN 0031      WRITE(6,960)
ISN 0032      GO TO 70
ISN 0033      40 WRITE(6,940)
ISN 0034      WRITE(6,970)
ISN 0035      RETURN
ISN 0036      50 WRITE(6,960)
ISN 0037      REWIND 11
ISN 0038      STOP 7777
ISN 0039      60 WRITE(6,970)
ISN 0040      C*****
ISN 0041      C BACKSPACE THE TAPE (OR DISK) TO THE BEGINNING OF THE DATA
ISN 0042      C RECORDS TO PREPARE FOR THE NEXT CASE.
ISN 0043      C*****
ISN 0044      REWIND 11
ISN 0045      DO 65 I=1,3
ISN 0046      65 READ(11)
ISN 0047      RETURN
ISN 0048      70 REWIND 11
ISN 0049      STOP 5555
ISN 0050      END
```

PROGRAM FOR PRINTING OUT STEADY-STATE SUMMARIES

55 of 87

PAGE 002

NAME	TAG	TYPE	ADD.
I		I*4	00000C
NCAT	C	I*4	N.R.
NTST	C	I*4	000028
NTFIN	C	I*4	00003C
ISCOM#	XF	I*4	000000

NAME	TAG	TYPE	ADD.
N		I*4	000010
NERR		I*4	000014
TFIN	C	R*4	N.R.
NWCAT	C	I*4	N.R.

NAME	TAG	TYPE	ADD.
TST	C	R*4	N.R.
NLIN	C	I*4	N.R.
NDNAM	C	I*4	N.R.
ITFIN	C	I*4	000054

NAME	TAG	TYPE	ADD.
ITST	C	I*4	000050
NSET	C	I*4	N.R.
NDUMP	C	I*4	N.R.
ERROR		R*4	N.R.

PROGRAM FOR PRINTING OUT STEADY-STATE SUMMARIES

56 of 87

PAGE 003

LABEL	ADDR	LABEL	ADDR
100000	18A821	100001	000336
100003	0003D4	100004	00040E
50	000490	60	0004BE
50	000490		

LABEL	ADDR	LABEL	ADDR
10	000350	20	000392
30	00041A	40	00045E
65	0004E8	100005	00050E

***** END OF COMPILATION *****

LEVEL 2 FEB 67

OS/360 FORTRAN H

DATE 67.236/12.40.35

COMPILER OPTIONS - NAME= MAIN,OPT=00,LINECNT=72,SOURCE,BCD,NOLIST,NODECK,LOAD,MAP,NGEDIT,ID

```

ISN 0002      SUBROUTINE HRCARD(NOP)
C*****
ISN 0003      REAL*8 DNAM,CDNAM,END
ISN 0004      COMMON/BLK1/DNAM(1100),CDNAM(17,200),COMENT(17,200,10)
ISN 0005      COMMON/BLK2/ NMAX(17), PDAT(5,200), LDAT(17,200),
1          DDAT(5,1100), LPSK(1100), KUNC, KCRD
ISN 0006      COMMON/BLK3/ TST(5), TFIN(5), NTST(5), NTFIN(5), ITST, ITFIN
ISN 0007      COMMON/BLK4/ NCAT(17), NLIN, NNCAT, NDNAM, NSET, NDUMP
ISN 0008      COMMON/BLK5/ DATE(2)
ISN 0009      COMMON/BLK6/CGYTTL(18,17)
ISN 0010      DIMENSION
1          CGY1 (18), CGY2 (18), CGY3 (18), CGY4 (18), CGY5 (18),
1          CGY6 (18), CGY7 (18), CGY8 (18), CGY9 (18), CGY10(18),
1          CGY11(18), CGY12(18), CGY13(18), CGY14(18), CGY15(18),
1          CGY16(18), CGY17(18)
ISN 0011      EQUIVALENCE
1          (CGYTTL(1, 1),CGY1 (1)), (CGYTTL(1, 2),CGY2 (1)),
1          (CGYTTL(1, 3),CGY3 (1)), (CGYTTL(1, 4),CGY4 (1)),
1          (CGYTTL(1, 5),CGY5 (1)), (CGYTTL(1, 6),CGY6 (1)),
1          (CGYTTL(1, 7),CGY7 (1)), (CGYTTL(1, 8),CGY8 (1)),
1          (CGYTTL(1, 9),CGY9 (1)), (CGYTTL(1,10),CGY10(1)),
1          (CGYTTL(1,11),CGY11(1)), (CGYTTL(1,12),CGY12(1)),
1          (CGYTTL(1,13),CGY13(1)), (CGYTTL(1,14),CGY14(1)),
1          (CGYTTL(1,15),CGY15(1)), (CGYTTL(1,16),CGY16(1)),
1          (CGYTTL(1,17),CGY17(1))
ISN 0012      DATA END/'END'
C*****
ISN 0013      900 FORMAT(65X,12,13,T1,A8,2X,10A4)
ISN 0014      910 FORMAT(17I2,35X,13,2A4)
ISN 0015      920 FORMAT(10I8)
C*****
ISN 0016      IF(NOP .EQ. 2) GO TO 20
C*****
C          READ THE CHANNEL NAMES (CDNAM), DESCRIPTIONS (COMENT) AND
C          CATEGORY NUMBER, I. DETERMINE THE MAXIMUM NUMBER OF CHANNELS
C          NMAX IN EACH CATEGORY.
C*****
ISN 0018      10 READ(5,900,END=130)I,J, CDNAM(I,J),(COMENT(I,J,K), K=1,10)
ISN 0019      IF(CDNAM(I,J) .EQ. END) GO TO 150
ISN 0021      IF(J .GT. NMAX(I)) NMAX(I) = J
ISN 0023      GO TO 10
C*****
C          READ THE REQUESTED CATEGORY NUMBERS NCAT AND THE TST DATE.
C          CHECK THAT CATEGORY 17, THE CRD CATEGORY IS PRESENT ALONE. IF
C          IT IS NOT, PRINT A MESSAGE AND GO TO THE NEXT CASE.
C*****
ISN 0024      20 READ(5,910, END=140)(NCAT(I), I=1,17), NDUMP, DATE
ISN 0025      IF(NCAT(1) .NE. 17) GO TO 40
ISN 0027      IF(NCAT(2) .EQ. 0) GO TO 60
ISN 0029      30 CALL ERROR(4)
ISN 0030      READ(5,910)
ISN 0031      GO TO 20
C*****
C          SCAN REMAINING NCAT FOR 17, THE CRD GROUP
C*****
ISN 0032      40 DO 50 I=2,17
ISN 0033      IF(NCAT(I) .EQ. 17) GO TO 30
ISN 0035      IF(NCAT(I) .EQ. 0) GO TO 70
ISN 0037      50 CONTINUE
ISN 0038      GO TO 70
ISN 0039      60 KCRD = 1
ISN 0040      GO TO 80
ISN 0041      70 KUNC = 1
ISN 0042      80 IF(KUNC .EQ. 0) GO TO 90
ISN 0044      IF(KCRD .EQ. 0) GO TO 100
ISN 0046      GO TO 30
ISN 0047      90 IF(KCRD .EQ. 0) GO TO 30
C*****
C          READ THE REQUESTED TIME INTERVALS OR SETS.
C*****
ISN 0049      100 READ(5,920,END=140)(NTST(I), NTFIN(I), I=1,5)

```

```

C*****
C    FIND THE NUMBER OF SFTS REQUESTED.
C*****
ISN 0050      NSET = 0
ISN 0051      DO 110 I=1,5
ISN 0052        IF(NTST(I) .NE. 0) NSET = NSET + 1
ISN 0054      110 CONTINUE
ISN 0055        IF(NSET .EQ. 0) GO TO 20
C*****
C    CONVERT THE START AND END TIMES TO DECIMAL.
C*****
ISN 0057      DO 120 I=1,5
ISN 0058        TST(I) = FLOAT(NTST(I))/10.
ISN 0059        TFIN(I) = FLOAT(NTFIN(I))/10.
ISN 0060      120 CONTINUE
ISN 0061      RETURN
ISN 0062      130 WRITE(6,5000)
ISN 0063        WRITE(6,5020)((I,J,CDNAM(I,J),J=1,200),I=1,17)
ISN 0064        WRITE(6,5030)((COMENT(I,J,K),K=1,10),J=1,200),I=1,17)
ISN 0065        STOP 9999
ISN 0066      140 CALL ERROR(5)
ISN 0067      5000 FORMAT(' HUNG UP IN CDNAM RECORDS.')
ISN 0068      5010 FORMAT(10X,A8,10X,10A4,2I10)
ISN 0069      5020 FORMAT(2X,2I10,10X,A8)
ISN 0070      5030 FORMAT(5(10X,10A4))
ISN 0071      5040 FORMAT('1'27X'----- REQUESTED CHANNEL TABULATION -----')
ISN 0072      5050 FORMAT('0',20X,18A4)
ISN 0073      5060 FORMAT('+',71X,'(CONTINUED)')
ISN 0074      5070 FORMAT(/74X'SYSTEM'4X,'SYSTEM'/10X,'CHANNEL'16X,'DESCRIPTION'
        1 16X,'UNITS'11X,'NO.'6X,'ITEM'/)
        5080 FORMAT(/7X,'----- REQUESTED CHANNEL SUMMARY -----'/7X,
        1  'SYSTEM NUMBER'13,16I5//4X,'CHANNELS/SYSTEM'14,16I5//40X,
        2  'TOTAL CHANNELS REQUESTED =',I5)
ISN 0075      5090 FORMAT(' ')
ISN 0076      150 LIN = 0
ISN 0077        KPG1 = 1
ISN 0078        DO 220 I=1,17
ISN 0079          NEWI = 1
ISN 0080          NINCAT = NMAX(I)
ISN 0081          DO 220 J=1,NINCAT
ISN 0082            IF(LIN .NE. 0) GO TO 180
ISN 0083            WRITE(6,5040)
ISN 0084            IF(KPG1 .EQ. 1) GO TO 160
ISN 0085            WRITE(6,5060)
ISN 0086            GO TO 170
ISN 0087          160 KPG1 = 0
ISN 0088          170 WRITE(6,5070)
ISN 0089            WRITE(6,5050) (CGYTTL(K,I),K=1,18)
ISN 0090            LIN = LIN + 2
ISN 0091            NEWI = 0
ISN 0092          180 IF(NEWI .NE. 1) GO TO 190
ISN 0093            WRITE(6,5050) (CGYTTL(K,I),K=1,18)
ISN 0094            LIN = LIN + 2
ISN 0095            NEWI = 0
ISN 0096          190 IF(NMAX(I) .EQ. 0) GO TO 200
ISN 0097            WRITE(6,5010) CDNAM(I,J),(COMENT(I,J,K),K=1,10),I,J
ISN 0100            GO TO 210
ISN 0101          200 WRITE(6,5090)
ISN 0102          210 LIN = LIN + 1
ISN 0103            IF(LIN .GT. 72) LIN = 0
ISN 0104          220 CONTINUE
ISN 0105            NCHSUM = 0
ISN 0106            DO 230 I=1,17
ISN 0107              NCHSUM = NCHSUM + NMAX(I)
ISN 0108            230 WRITE(6,5080) (I,I=1,17), (NMAX(I),I=1,17),NCHSUM
ISN 0109            RETURN
ISN 0110            END
ISN 0111
ISN 0112
ISN 0113

```

PROGRAM FOR PRINTING OUT STEADY-STATE SUMMARIES

59 of 87

PAGE 003

NAME	TAG	TYPE	ADD.
I		I**	000044
LIN		I**	000050
CGY2	CE	R**	N.R.
CGY6	CE	R**	N.R.
DATE	C	R**	000000
KCRD	C	I**	00AC28
LDSK	C	I**	N.R.
NMAX	C	I**	000000
TFIN	C	R**	000014
NWCAT	C	I**	N.R.
CGY10	CL	R**	N.R.
CGY14	CF	R**	N.R.
IBCOM#	XF	I**	000000
COMENT	C	R**	008CA0

NAME	TAG	TYPE	ADD.
J		I**	000048
NUP		I**	000054
CGY3	CE	R**	N.R.
CGY7	CE	R**	N.R.
DDAT	C	R**	N.R.
KPG1		I**	000058
NCAT	C	I**	000000
NSET	C	I**	000050
NDNAM	C	I**	N.R.
ITFIN	C	I**	N.R.
CGY11	CE	R**	N.R.
CGY15	CE	R**	N.R.
RDCARD		R**	N.R.
CGYTTL	CE	R**	000000

RDCARD /

NAME	TAG	TYPE	ADD.
K		I**	00004C
TST	C	R**	00C000
CGY4	CE	R**	N.R.
CGY8	CF	R**	N.R.
DNAM	C	R**	N.R.
KUNC	C	I**	00AC24
NCWI		I**	00005C
NTST	C	I**	000028
NDUMP	C	I**	000054
ERROR	XF	R**	000000
CGY12	CE	R**	N.R.
CGY16	CE	R**	N.R.
NINCAT		I**	000060

NAME	TAG	TYPE	ADD.
END		R**	000068
CGY1	CE	R**	N.R.
CGY5	CE	R**	N.R.
CGY9	CE	R**	N.R.
ITST	C	I**	N.R.
LDAT	C	I**	N.R.
NLIN	C	I**	N.R.
PDAT	C	R**	N.R.
NTFIN	C	I**	00003C
CDNAM	C	R**	002260
CGY13	CE	R**	N.R.
CGY17	CE	R**	N.R.
NCHSUM		I**	000064

PROGRAM FOR PRINTING OUT STEADY-STATE SUMMARIES

60 of 87

PAGE 004

LABEL	ADDR	LABEL	ADDR	LABEL	ADDR	LABEL	ADDR
100000	000000	100001	000486	100010	000494	100002	0004E8
100003	000536	100004	000566	100005	000580	100006	000594
20	00059A	100007	000588	100008	0005E2	100009	00060E
30	00061E	40	000646	100010	00064E	100011	000668
50	000680	100012	000696	60	00069C	70	0006AE
80	00068A	100013	0006CA	100014	0006DA	90	0006E0
100	0006F0	100015	00070C	100016	000746	100017	00075E
100018	000776	110	000786	100019	00079C	100020	0007AC
100021	000784	120	000802	100022	000866	130	00086E
100023	00089C	100024	0008A4	100025	0008F2	100026	000908
100027	000924	100028	00092C	100029	000934	100030	000982
100031	000998	100032	0009AE	140	0009C0	150	0009CE
100033	0009E2	100034	000A06	100035	000A12	100036	000A36
160	000A52	170	000A5A	100037	000A88	100038	000AC6
180	000ADE	100039	000AEC	100040	000B04	100041	000B42
190	000B5A	100042	000B72	100043	000BB4	100044	000C02
200	000C1E	210	000C34	100045	000C4A	220	000C52
100046	000C68	100047	000C7E	230	000C8E	100048	000CBC
100049	000CD4	100050	000CF2	100051	000CFA	100052	000D26

***** END OF COMPILATION *****

PROGRAM FOR PRINTING OUT STEADY-STATE SUMMARIES

61 of 87

LEVEL 2 FEB 67

OS/360 FORTRAN H

DATE 67.236/12.40.43

COMPILER OPTIONS - NAME= MAIN.OPT=00.LINECNT=72.SOURCE,BCD,NOLIST,NODECK,LOAD,MAP,NOEDIT,ID

PROGRAM FOR PRINTING OUT STEADY-STATE SUMMARIES

62 of 87

```

IEF285I      SYSOUT                      SYSOUT
IEF285I      VOL SER NOS= T07426.
IEF285I      SYS1.UT4                      KEPT
IEF285I      VOL SER NOS= D10001.          KEPT
IEF285I      PCHOUT
IEF285I      VOL SER NOS= D10012.
IEF285I      AAAAAAAAA.AAAAAAAAA.AAAAAAAAA.00000006 SYSIN
IEF285I      VOL SER NOS= SYSIN .
//PUNCH      EXEC PGM=PUNCHCDS             00000060
//PIN        DD      DSN=PCCHOUT,DISP=OLD   00000070
//PQUT       DD      SYSOUT=A               00000080
IEF236I      ALLOC. FOR 057105 PUNCH STEPNAME
IEF237I      PIN      ON 132
    
```

PROGRAM FOR PRINTING OUT STEADY-STATE SUMMARIES

63 of 87

```

IEF285I PCHOUT
IEF285I VOL SER NOS= D10012.
//LINK EXEC PGM=IEWL,PARM='XREF,LET,LIST'
//SYSLIB DD DSN=SYS1.FORTLIB, SYSTEM SUBROUTINES 00000090
// DISP=OLD 100000100
// DD DSN=SYS1.ASLIB,DISP=OLD 00000110
//SYSLMOD DD DSN=LOADMOD(MAIN), 00000120
// DISP=(NEW,PASS),UNIT=DISKUTIL, XC0000130
// SPACE=(3350,(660,100,20)) 200000140
//SYSD1 DD DSN=SYS1.UT1,DISP=OLD 00000150
//SYSPRINT DD SYSOUT=A 00000160
//LOADM DD DSN=CHKCLMOD,DISP=OLD 00000170
// DD DSN=CHKOLMD2,DISP=OLD 00000180
//SYSLIN DD DSN=*.COMPIL.SYSLIN,DISP=OLD,DCB=*.COMPIL.SYSLIN 00000190
// DD DDNAME=XLOADM 00000200
// CONCATENATE CARD READR500000210
// DECKS (OBJ) IF NEEDED600000220
// HERE WITH DEEDEE STMT 00000230

IEF236I ALLOC. FOR D57105 LINK STEPNAME
IEF237I SYSLIB ON 130
IEF237I ON 132
IEF237I SYSLMOD ON 133
IEF237I SYSD1 ON 131
IEF237I LOADM ON 133
IEF237I ON 133
IEF237I SYSLIN ON 130

```

F--LEVEL LINKAGE EDITOR OPTIONS SPECIFIED--XREF,LET,LIST
IEW0000 MAIN NOW ADDED TO DATA SET

---- CROSS REFERENCE TABLE ----

CONTROL SECTION			ENTRY							
NAME	ORIGIN	LENGTH	NAME	LOCATION	NAME	LOCATION	NAME	LOCATION	NAME	LOCATION
IHCFCOMH*	00	FFD	IBCOM#	00	FDIOCS#	8C				
IHCUCOPT *	1000	8								
IHCFCVTH*	1008	FF3	ADCON#	1008	FCVZO	1154	FCVAO	11FA	FCVLO	1282
			FCVIO	1590	FCVEO	1A34	FCVCO	1C2E		
IHCTRCH *	2000	258								
BLK1	2258	29FE0	MAIN	37368						
BLK2	2C238	AC2C	RDISK	37768						
BLK6	36E68	4C8	DXFER	37EF0						
MAIN#	37330	270	PRTDAT	38298						
RDISK#	375A0	926	NAMCEK	38830						
DXFER#	37EC8	21C	ERROR	38D20						
PRTDAT#	380E8	680	RDCARD	39238						
NAMCEK#	38768	388	FIOCS#	39D68						
ERROR#	38AF0	532								
RDCARD#	39028	D3C								
IHCFIOSH*	39D68	CF2								
IHCUATBL*	3AA60	638								
BLK3	38098	58								
BLK4	380F0	58								
BLK5	38148	8								

LOCATION REFERS TO SYMBOL IN CONTROL SECTION

EA8	ADCON#	IHCFCVTH
EA0	FIOCS#	IHCFIOSH
DB0	IHCUCOPT	IHCUCOPT
EAC	FCVEO	IHCFCVTH
EB0	FCVLO	IHCFCVTH
EB4	FCVIO	IHCFCVTH
EB8	FCVCO	IHCFCVTH
EBC	FCVAO	IHCFCVTH
EC0	FCVZO	IHCFCVTH
E98	IHCTRCH	IHCTRCH

LOCATION	REFERS TO SYMBOL	IN CONTROL SECTION
2158	IBCOM#	IHCFCOMH
215C	ADCON#	IHCFCVTH
2160	FIOCS#	IHCFIOSH
1E8C	IHC0M#	IHCFCOMH
3743E	BLK1	BLK1
3743C	BLK1	BLK1
37440	BLK1	BLK1
37444	BLK2	BLK2
37448	BLK2	BLK2
3744C	BLK2	BLK2
37450	BLK2	BLK2
37454	BLK3	BLK3
37458	BLK4	BLK4
3745C	BLK5	BLK5
37460	BLK6	BLK6
37464	RODISK	RODISK#
37468	ERRDR	ERROR#
3746C	DXFER	DXFER#
37470	RDCARD	RDCARD#
37474	PRTDAT	PRTDAT#
37478	IBCOM#	IHCFCOMH
3742C	BLK4	BLK4
37814	BLK1	BLK1
37818	BLK1	BLK1
3781C	BLK1	BLK1
37820	BLK2	BLK2
37824	BLK2	BLK2
37828	BLK2	BLK2
3782C	BLK2	BLK2
37830	BLK3	BLK3
37834	BLK4	BLK4
37838	ERROR	ERROR#
3783C	IBCOM#	IHCFCOMH
37840	NAMCEK	NAMCEK#
37E24	BLK1	BLK1
37E68	BLK2	BLK2
37EAC	BLK2	BLK2
37F48	BLK2	BLK2
37F4C	BLK2	BLK2
37F50	BLK2	BLK2
37F54	BLK2	BLK2
382F0	BLK1	BLK1
382F4	BLK1	BLK1
382F8	BLK1	BLK1
382FC	BLK2	BLK2
38300	BLK2	BLK2
38304	BLK2	BLK2
38308	BLK2	BLK2
3830C	BLK3	BLK3
38310	BLK4	BLK4
38314	BLK5	BLK5
38318	BLK6	BLK6

LOCATION	REFERS TO SYMBOL	IN CONTROL SECTION
3831C	IBCOM#	IHCFCOMH
38428	BLK5	BLK5
38078	BLK3	BLK3
3807C	BLK4	BLK4
38080	IBCOM#	IHCFCOMH
38888	BLK1	BLK1
3888C	BLK1	BLK1
38890	BLK1	BLK1
38894	BLK2	BLK2
38898	BLK2	BLK2
3889C	BLK2	BLK2
388A0	BLK2	BLK2
388A4	BLK4	BLK4
388A8	IBCOM#	IHCFCOMH
39208	BLK1	BLK1
3920C	BLK1	BLK1
392E0	BLK1	BLK1
392E4	BLK2	BLK2
392E8	BLK2	BLK2
392EC	BLK2	BLK2
392F0	BLK2	BLK2
392F4	BLK3	BLK3
392F8	BLK4	BLK4
392FC	BLK5	BLK5
39300	BLK6	BLK6
39304	ERROR	ERROR#
39308	IBCOM#	IHCFCOMH
3961C	BLK5	BLK5
39E90	IHCUATBL	IHCUATBL
39E9C	IBCOM#	IHCFCOMH
ENTRY ADDRESS	37368	
TOTAL LENGTH	38150	

```

IEF285I  SYS1.FORTLIB          KEPT
IEF285I  VOL SER NOS= D10001.
IEF285I  SYS1.ASLIB           KEPT
IEF285I  VOL SER NOS= C10012.
IEF285I  LOADMOD.D57105      PASSED
IEF285I  VOL SER NOS= D10011.
IEF285I  SYS1.UT1            KEPT
IEF285I  VOL SER NOS= D10002.
IEF285I  SYSJUT              SYSOUT
IEF285I  VOL SER NOS= T07426.
IEF285I  CHKOLMOD            KEPT
IEF285I  VOL SER NOS= D10011.
IEF285I  CHKOLMOD2          KEPT
IEF285I  VOL SER NOS= D10011.
IEF285I  SYS1.UT4            KEPT
IEF285I  VOL SER NOS= D10001.
//PERF    EXEC PGM=*.LINK.SYSLMOD      00000240
//SYSABEND DD SYSOUT=A                00000250
//FT06F001 DD SYSOUT=A,DCB=(RECFM=VBA,LRECL=138,BLKSIZE=730) 00000260
//FT07F001 DD DSNAME=PCMCOUT,DISP=(OLD,KEEP) 00000270
//FT01F001 DD UNIT=DISKUTIL,DSNAME=ERS1, 100000280
//          SPACE=(800,(40,20)),DISP=NEW 00000290
//FT01F002 DD DSNAME=*.FT01F001,VOLUME=REF=*.FT01F001, 100000300
//          DISP=(OLD,DELETE) 00000310
//FT02F001 DD UNIT=DISKUTIL,DSNAME=ERS2,SEP=(FT01F001), 100000320
//          SPACE=(800,(40,20)),DISP=NEW 00000330
//FT03F001 DD UNIT=DISKUTIL,DSNAME=ERS3,SEP=(FT01F001,FT02F001), 100000340
//          SPACE=(800,(40,20)),DISP=NEW 00000350
//FT02F002 DD DSNAME=*.FT02F001,VOLUME=REF=*.FT02F001, 100000360
//          DISP=(OLD,DELETE) 00000370
//FT03F002 DD DSNAME=*.FT03F001,VOLUME=REF=*.FT03F001, 100000380
//          DISP=(OLD,DELETE) 00000390
//PERF.FT11F001 DD UNIT=TMNTG,DSNAME=TESDAT,DISP=(OLD,KEEP), 1
//          DCB=(DEN=2,RECFM=VB,LRECL=3604,BLKSIZE=7212,BUFNO=2), 2
//          VOLUME=SER=T05608
//PERF.FT05F001 DD *
IEF236I  ALLOC. FOR D57105  PERF  STEPNAME
IEF237I  PGM=*.DD ON 133
IEF237I  FT07F001 ON 132
IEF237I  FT01F001 ON 130
IEF237I  FT01F002 ON 130
IEF237I  FT02F001 ON 134
IEF237I  FT03F001 ON 135
IEF237I  FT02F002 ON 134
IEF237I  FT03F002 ON 135
IEF237I  FT11F001 ON 185
IEF237I  FT05F001 ON 182

```

PROGRAM FOR GENERATING MULTIPLE PLOTS
(Step 1: Generation of Disk Data Sets for Each Time Interval)

68 of 87

```

FORTRAN IV G LEVEL 0, MOD 0      MAIN      DATE = 67215      20/12/47      PAGE 0001

C
C      SUPPLEMENTARY MAIN PROGRAM FOR STEP1 TO READ DATA TAPE CONTAINING
C      ALL PLOT CHANNELS FOR EACH TIME STEP
C
C      A(I) = CHANNEL NAMES IN ORDER PRESENT ON TAPE AT EACH TIME
C      AORIG(I) = UNORDERED CHANNEL NAMES
C      IDUM(I) = DUMMY STORAGE
C      NV = NUMBER OF CHANNELS ON TAPE
C      Y(I) = VALUE OF CHANNEL I AT CURRENT TIME
C      LP(I) = LOCATION OF UNORDERED CHANNEL I IN ORDERED CHANNEL VALUES
C      AT EACH TIME
C      TIME = CURRENT TIME (.1 SEC)
C
0001      REAL*8 A(1100), AORIG(1100)
0002      DIMENSION IDUM(23), Y(1100), LP(1100)
0003      DIMENSION ASINW(2200)
0004      EQUIVALENCE (ASINW(1),AORIG(1))
C
C      READ FIRST RECORD CONTAINING NUMBER OF CHANNELS
C
0005      READ (12)(IDUM(I),I=1,23),NV
0006      WRITE(6,9000)
0007      9000 FORMAT(2H 1////////)
0008      NV2=NV*2
C
C      READ SECOND RECORD OF CHANNEL NAMES (UNORDERED)
C
0009      READ (12) IDUM(I),(ASINW(I),I=1,NV2)
0010      WRITE(6,9001)
0011      9001 FORMAT(2H 2////////)
C
C      READ THIRD RECORD OF LOCATION OF UNORDERED CHANNELS
C
0012      READ (12)IDUM(I),(LP(I),I=1,NV)
0013      WRITE(6,9002)
0014      9002 FORMAT(2H 3////////)
C
C      ORDER CHANNEL NAMES
C
0015      DO 100 J=1,NV
0016      A(LP(J)-3)=AORIG(J)
0017      100 CONTINUE
C
C      INITIALIZE ROUTINE TO SELECT CHANNELS TO BE PLOTTED
C
0018      CALL WNAME(A,NV)
0019      WRITE(6,9003)
0020      9003 FORMAT(2H 4////////)
C
C      READ EACH TIME RECORD AND PASS CHANNEL DATA TO SELECTION SUBROUT.
C
0021      150 READ (12,END=200)(IDUM(J),J=1,3),(Y(I),I=1,NV)
0022      CALL WDISK(IDUM(2),Y)
0023      GO TO 150
C
C      TERMINATE CHANNEL DATA SELECTION
C
0024      200 CALL WMAX(IDUM(2))
0025      WRITE(6,9004)
0026      9004 FORMAT(2H 5////////)
0027      STOP
0028      END

```


PAGE 0002

TOTAL MEMORY REQUIREMENTS 006BEE BYTES

FORTRAN IV G LEVEL 0, MOD 0

WNAME

DATE = 67215

20/12/47

PAGE 0001

```

0001      SUBROUTINE WNAME(A,NV)
C
C      READ INPUT FOR PLOTTING THE PLOT DATA TAPE FOR W74740
C
C      A(I) = CHANNEL NAME FOR I-TH DATA CHANNEL VALUE OF ARRAY Y, #HERE
C      Y IS AN ARGUMENT OF WDISK
C      NV = NUMBER OF CHANNELS SUPPLIED AT EACH TIME BY CALLING PROGRAM
C      (MAX OF 1100)
C      INDA(K,N) = FOR TIME INTERVAL N, THE INDEX IN THE A ARRAY OF THE
C      K-TH UNIQUE CHANNEL REQUESTED FOR PLOTTING (IN THE
C      ORDER OF REQUEST AND THUS MULTIPLY INPUT)
C
C      NTIME = NUMBER OF TIME INTERVALS (MAX OF 4)
C      TMIN(N) = MINIMUM TIME OF INTERVAL N
C      (.1 SEC)
C      TMAX(N) = MAXIMUM TIME OF INTERVAL N
C      NMPLT(N) = NUMBER OF MULTIPLOTS FOR TIME INTERVAL N (MAX OF 500)
C      JYSCL = NUMBER OF VERTICAL SCALES FOR MULTIPLY M OF TIME
C      INTERVAL N (MAX OF 3)
C      YMIN = MINIMUM VERTICAL SCALE VALUE FOR SCALE J OF
C      MULTIPLY M
C      YMAX = MAXIMUM VERTICAL SCALE VALUE
C      KGROUP = NUMBER OF GROUPS USING SCALE J OF MULTIPLY M OF TIME
C      INTERVAL N
C      (MAX OF 10 --- SEE NOTE FOR FURTHER RESTRICTIONS)
C      NCHAN = NUMBER OF CHANNELS (SEE NOTE ON RESTRICTIONS)
C      CHAN(IKJ) = CHANNEL NAME I FOR GROUP K, SCALE J, MULTIPLY M,
C      TIME INTERVAL N. THE CHANNELS FOR EACH MULTIPLY
C      ARE PACKED CONSECUTIVELY IN STORAGE.
C      *NOTE* - SUM OF NCHAN VALUES FOR A SINGLE MULTIPLY (ALL GROUPS
C      AND VERTICAL SCALES) HAS A MAX. OF 32
C      *NOTE* - FOR ANY VERTICAL SCALE, SUM OF NCHAN VALUES AND 5*KGROUP
C      HAS A MAX. OF 66
C      NUM = NUMBER OF TIME VALUES FOR TIME INTERVAL (MAX OF 500)
C      DT(N) = TIME STEP FOR TIME INTERVAL N (.1 SEC)
C      NCTOT(N) = NUMBER OF CHANNELS USED FOR PLOTS FOR TIME INTERVAL N
C
0002      DIMENSION TMIN(4),TMAX(4), DT(4), NCTOT(4)
0003      DIMENSION TNEXT(4),IORD(4)
0004      DIMENSION ICOUNT(4)
0005      INTEGER*2 INDA(1100,4)
0006      REAL*8 A(1100), CHAN(32)
C
C      READ INPUT FOR EACH TIME INTERVAL
C      SAVE INFORMATION NEEDED FOR DISK STORAGE ONLY
C
0007      READ (5,1000) NTIME
0008      IF (NTIME.GT.4) GO TO 255
0009      DO 250 N=1,NTIME
C
C      INPUT MIN TIME AND MAX TIME FOR TIME INTERVAL
C
0010      READ(5,1050) I1,I2
0011      TMIN(N)=I1
0012      TMAX(N)=I2
0013      IF (TMIN(N).GT.TMAX(N)) GO TO 260
C
C      DETERMINE TIME STEP USED FOR TIME INTERVAL N
C
0014      L=1
0015      DTMM=TMAX(N)-TMIN(N)
0016      10 NUM=DTMM/FLGAT(L)+.01
0017      IF (NUM.LE.499) GO TO 20
0018      L=L+1
0019      GO TO 10
0020      20 DT(N)=FLOAT(L)
0021      NUM=NUM+1
C
C      READ MULTIPLY NUMBER FOR TIME INTERVAL
C

```

```

FORTRAN IV G LEVEL 0, MOD 0          WNAME          DATE = 67215          20/12/47          PAGE 0002

0022      READ (5,1000) MPLOTT
0023      IF( MPLOTT.GT.500)GO TO 265
      C
      C      READ CARDS 5-10 FOR EACH MULTIPLY
      C
0024      NCT=0
0025      DO 225 M=1,MPLOTT
0026      NC=0
0027      READ (5,1030)DUM
0028      READ (5,1000)JYSCL
0029      IF(JYSCL.GT.3)GO TO 270
      C
      C      READ CARDS 7-10 FOR EACH VERTICAL SCALE
      C
0030      DO 110 J=1,JYSCL
0031      READ (5,1010)DUM,DUM,KGROUP
0032      IF(KGROUP.GT.10)GO TO 275
      C
      C      READ CARDS 8-10 FOR EACH GROUP
      C
0033      NLINE=5*KGROUP
0034      DO 100 K=1,KGROUP
0035      READ (5,1000)NCHAN
0036      NCKM=NC+1
0037      NC=NC+NCHAN
0038      IF(NC.GT.32)GO TO 280
0039      NLINE=NLINE+NCHAN
0040      IF(NLINE.GT.66)GO TO 285
0041      READ (5,1020)(CHAN(I),I=NCKM,NC)
0042      READ (5,1030)DUM
0043      100 CONTINUE
0044      110 CONTINUE
      C
      C      GENERATE ARRAY OF CHANNEL INDICES FROM INPUT FOR EACH PLOT
      C
      C      FIND INPUT CHANNEL NAME IN ARRAY A
      C
0045      DO 200 I=1,NC
0046      DO 120 J=1,NV
0047      IF(CHAN(I).EQ.A(J))GO TO 140
0048      120 CONTINUE
0049      GO TO 290
      C
      C      TEST IF CHANNEL PREVIOUSLY REQUESTED
      C
0050      140 IF(NCT.EQ.0)GO TO 180
0051      DO 160 K=1,NCT
0052      IF(J.EQ.INDA(K,N))GO TO 200
0053      160 CONTINUE
0054      180 NCT=NCT+1
0055      INDA(NCT,N)=J
0056      200 CONTINUE
0057      225 CONTINUE
0058      NCTOT(N)=NCT
      C
      C      INITIAL WRITE ON DISK FOR TIME INTERVAL N
      C
      C      RECORD 1 - NUMBER OF CHANNELS, NUMBER OF TIME VALUES
      C
      C      RECORD 2,3 - CHANNEL NAMES A(J)
      C
0059      NW=N
0060      WRITE (NW)NCTOT(N),NUM
0061      IF(NCT.GT.904)GO TO 235
0062      WRITE (NW) (A(INDA(NC,N)),NC=1,NCT)
0063      GO TO 250
0064      235 WRITE (NW) (A(INDA(NC,N)),NC=1,904)
0065      WRITE (NW)(A(INDA(NC,N)),NC=905,NCT)
0066      250 CONTINUE
0067      GO TO 320
      C
      C      ERROR MESSAGE S

```

FORTRAN IV G LEVEL 0, MOD 0

WNAME

DATE = 67215

20/12/47

PAGE 0003

```

C
0068      255 WRITE (6,2980)NTIME
0069      STOP
0070      260 WRITE (6,2990)N,TMIN(N),TMAX(N)
0071      STOP
0072      265 WRITE (6,3000)N,M,PLOT
0073      STOP
0074      270 WRITE (6,3010)N,M,J,YSCAL
0075      STOP
0076      275 WRITE (6,3020)N,M,J,K,GROUP
0077      STOP
0078      280 WRITE (6,3030)N,M,J,K,NC
0079      STOP
0080      285 WRITE (6,3040)N,M,J,K,NLINE
0081      STOP
0082      290 WRITE (6,3050)N,M,CHAN(I)
0083      STOP

C
C      FORMAT STATEMENTS
C
0084      1000 FORMAT(12I6)
0085      1010 FORMAT(2E12.5,I6)
0086      1020 FORMAT(6(A8.4X))
0087      1030 FORMAT(18A4)
0088      1050 FORMAT(6I12)
0089      2980 FORMAT(1H0/65H THE MAXIMUM NUMBER OF TIME INTERVALS IS 4. THE IN
0090      2990 FORMAT(1H0/18H FOR TIME INTERVAL ,I3,Ix,51H THE MINIMUM TIME IS GR
0091      3000 FORMAT(1H0/18H FOR TIME INTERVAL ,I3,Ix,61H THE MAXIMUM NUMBER OF M
0092      3010 FORMAT(1H0/63H THE MAXIMUM NUMBER OF VERTICAL SCALES FOR EACH MULT
0093      3020 FORMAT(1H0/60H THE MAXIMUM NUMBER OF GROUPS FOR EACH VERTICAL SCAL
0094      3030 FORMAT(1H0/57H THE MAXIMUM NUMBER OF CHANNELS FOR EACH MULTIPLY I
0095      3040 FORMAT(1H0/ 79H THE MAXIMUM VALUE OF THE SUM OF (NO. OF CHANNELS)
0096      3050 FORMAT(1H0/18H FOR TIME INTERVAL ,I2,Ix,10H,MULTIPLY ,I4,Ix,13H,T
C
C      INITIALIZE VARIABLES FOR WDISK
C
C      TNEXT(N) = NEXT TIME USED FOR PLOTS OF TIME INTERVAL N
C      IORD(N)= INDICES FOR TIME INTERVALS ORDERED BY ASCENDING TMIN(N)
C      NORD = NUMBER OF VALUES CURRENTLY IN IORD ARRAY
C      (NUMBER OF TIME INTERVALS TO COMPLETE DATA STORAGE FOR)
C      NN = CURRENT TIME INTERVAL BEING CONSIDERED
0097      320 CONTINUE
0098      DO 330 N=1,NTIME
0099      TNEXT(N)=TMIN(N)
0100      IORD(N)=N
0101      ICOUNT(N)=0
0102      330 CONTINUE
0103      NTIMM=NTIME-1
0104      IF(NTIMM.EQ.0) GO TO 395
0105      DO 390 N=1,NTIMM
0106      I=IORD(N)
0107      NP=N+1
0108      DO 360 N2=NP,NTIME

```

```

FORTRAN IV G LEVEL 0, MOD 0          WNAME          DATE = 67215          20/12/47          PAGE 0004

01C9          I1=IORD(N2)
0110          IF(TMIN(I).LE.TMIN(I1))GO TC 360
0111          I=I1
0112          360 CONTINUE
0113          IORD(N)=I
0114          IORD(I)=N
0115          390 CONTINUE
0116          395 CONTINUE
0117          NORD=NTIME
0118          RETURN

C
0119          ENTRY WDISK(itime,y)
0120          TIME=itime

C
C      TEST TO DETERMINE INTERVALS USING CURRENT TIME
C      STORE PLOT DATA FOR TIME INTERVALS USING CURRENT TIME
C      WRITE RECORD ON DISK FOR TIME INTERVAL NN
C
C      TIME = CURRENT TAPE TIME
C      Y(I) = CURRENT CHANNEL DATA
C      INORD = INDEX IN ORDERED TIME INTERVAL ARRAY FOR CURRENT INTERVAL
C
0121          DIMENSION Y(1100)
C
C      TEST IF DISK DATA SETS COMPLETE FOR EACH TIME INTERVAL
C
C      IF(NORD.EQ.0)RETURN
0122          INORD=1
0123          NN=IORD(INORD)
0124          400 NN=IORD(INORD)
0125          IF(TIME.LT.TMIN(NN))RETURN
0126          IF(TIME.LT.TNEXT(NN))GO TO 600

C
C      WRITE SELECTED DATA ON DISK (CORRESPONDING TO TIME INTERVAL) WITH
C      ONE RECORD OF PARAMETER VALUES FOR A GIVEN TIME FORMING A RECORD
C
C
0127          NC=NC TOT(NN)
0128          NW=NN
0129          WRITE (NW)TIME,( Y(INDA(I,NN)),I=1,NC)
0130          460 CONTINUE

C
C      CHECK IF PLOTS FOR TIME INTERVAL COMPLETED
C
C
0131          IF(TIME.GE.TMAX(NN))GO TO 700
C
C      INCREMENT NEXT-TIME ARRAY
C
C
0132          500 CONTINUE
0133          TNEXT(NN)=TNEXT(NN)+DT(NN)
0134          IF(TIME.LT.TNEXT(NN))GO TO 600
0135          ICOUNT(NN)=1
0136          TNEX=TNEXT(NN)-DT(NN)
0137          WRITE (6,6020)TNEX,TMIN(NN),TMAX(NN)
0138          6020 FORMAT(1H0/ 9H0THE TIME ,E11.4 ,1X,43H(.1 SEC) WAS NOT INCLUDED 0
IN THE DATA TAPE. /51H THIS TIME HAS BEEN DELETED FROM TIME INTERVA
2L FROM ,E11.4 ,2HTO ,E11.4)
0139          GO TO 500

C
C      IF ADDITIONAL TIME INTERVALS EXIST, CHECK NEXT INTERVAL
C      IF NOT, RETURN TO CALLING PROGRAM
C
C
0140          600 CONTINUE
0141          IF(NORD.EQ.INORD)RETURN
0142          INORD=INORD+1
0143          GO TO 400

C
C      PLOT DATA STORAGE COMPLETE FOR TIME INTERVAL
C      ELIMINATE TIME INTERVAL FROM ORDERED INTERVAL ARRAY AND CONTINUE
C
C
0144          700 CONTINUE
0145          NORD=NORD-1

```

```

FORTRAN IV G LEVEL 0, MOD 0          WNAME          DATE = 67215          20/12/47          PAGE 0005

0146      IF(NORD.LT.INORD) RETURN
0147      DO 800 I=INORD,NORD
0148      IORD(I)=IORD(I+1)
0149      800 CONTINUE
0150      GOTO 400

C
0151      ENTRY WMAX(ETIME)
0152      TIME=ETIME

C
C      TEST IF ALL DATA SETS COMPLETE
C      IF NOT, WRITE LIST OF INCOMPLETE TIME INTERVALS
C
0153      IF(NORD.NE.0) WRITE(6,6000)TIME
0154      6000 FORMAT(1H0/ 43H0THE MAXIMUM TIME SUPPLIED FOR PLOTTING WAS, F6.0,
11X, 8H(.1 SEC) /68H0THE FOLLOWING TIME INTERVALS REQUESTED FOR PLO
2TTING WERE INCOMPLETE / 1H0,2X,4HTIME , 11X,4HTIME , 17X,4HTIME /
39H INTERVAL ,7X,7HMINIMUM ,14X,7HMAXIMUM /7H INDEX ,8X,8H(.1 SEC)
4 ,5X,9HREQUESTED ,8X,6HACTUAL )

0155      DO 900 NN=1,NTIME
0156      IF(NORD.EQ.0)GO TO 820
0157      DO 810 K=1,NORD
0158      IF(IORD(K).NE.NN)GO TO 810
0159      TNEXT(NN)=TNEXT(NN)-DT(NN)
0160      WRITE (6,6010)NN,TMIN(NN),TMAX(NN),TNEXT(NN)
0161      6010 FORMAT(1H ,14,4X,3F14.0)
0162      GO TO 830
0163      810 CONTINUE
0164      820 IF(ICOUNT(NN).EQ.0)GO TO 900
0165      830 CONTINUE

C
C      WRITE DUMMY RECORD WITH TIME=-TIME FOR INCOMPLETE TIME INTERVALS
C
0166      NW=NN
0167      TIM=-TIME
0168      NC=NC TOT(NN)
0169      WRITE (NW)TIM, (Y(J),J=1,NC)
0170      900 CONTINUE
0171      RETURN
0172      END

```

75 of 87

Step 1

FURTRAN IV G LEVEL 0. MOD 0 WNAME DATE = 67215 20/12/47 PAGE 0006

SYMBOL		LOCATION		SYMBOL		LOCATION		SYMBOL		LOCATION		SYMBOL		LOCATION	
SCALAR MAP															
SYMBOL	LOCATION	SYMBOL	LOCATION	SYMBOL	LOCATION	SYMBOL	LOCATION	SYMBOL	LOCATION	SYMBOL	LOCATION	SYMBOL	LOCATION	SYMBOL	LOCATION
NTIME	264	N	268	I1	26C	I2	270	L	274	L	274	L	274	L	274
DTMM	27E	NUM	27C	MPLCT	280	NCT	284	M	288	M	288	M	288	M	288
NC	28C	DUM	290	JYSCL	294	J	298	KGROUP	29C	KGROUP	29C	KGROUP	29C	KGROUP	29C
NLINE	2A0	K	2A4	NCHAN	2A8	NCKM	2AC	I	2B0	I	2B0	I	2B0	I	2B0
NV	2B4	NW	2B8	NTIMM	2BC	NP	2C0	N2	2C4	N2	2C4	N2	2C4	N2	2C4
II	2C8	NORD	2CC	TIME	2D0	ITIME	2D4	INORD	2D8	INORD	2D8	INORD	2D8	INORD	2D8
NN	2DC	TNE X	2E0	TIM	2E4										
ARRAY MAP															
SYMBOL	LOCATION	SYMBOL	LOCATION	SYMBOL	LOCATION	SYMBOL	LOCATION	SYMBOL	LOCATION	SYMBOL	LOCATION	SYMBOL	LOCATION	SYMBOL	LOCATION
TMIN	2E8	TMAX	2F8	DT	308	NCTDT	318	TNEXT	328	TNEXT	328	TNEXT	328	TNEXT	328
IORD	338	ICOUNT	348	INDA	358	A	25B8	CHAN	25C0	CHAN	25C0	CHAN	25C0	CHAN	25C0
Y	26C0														
SUBPROGRAMS CALLED															
SYMBOL	LOCATION	SYMBOL	LOCATION	SYMBOL	LOCATION	SYMBOL	LOCATION	SYMBOL	LOCATION	SYMBOL	LOCATION	SYMBOL	LOCATION	SYMBOL	LOCATION
IBCOM#	26C4														
FORMAT STATEMENT MAP															
SYMBOL	LOCATION	SYMBOL	LOCATION	SYMBOL	LOCATION	SYMBOL	LOCATION	SYMBOL	LOCATION	SYMBOL	LOCATION	SYMBOL	LOCATION	SYMBOL	LOCATION
1000	26D0	1010	26D6	1020	26DF	1030	26E8	1050	26EE	1050	26EE	1050	26EE	1050	26EE
29E0	26F4	2990	2754	3000	27EE	3010	2862	3020	2900	3020	2900	3020	2900	3020	2900
3030	29AD	3040	2A6F	3050	2B32	6020	2BC1	6000	2C44	6000	2C44	6000	2C44	6000	2C44
6010	2D38														

TOTAL MEMORY REQUIREMENTS 003B7C BYTES

FORTRAN IV G LEVEL 0, MOD 0

MAIN

DATE = 67216

22/08/16

PAGE 0001

```

C
C REORDER CHANNEL DATA ON DISK FOR SINGLE TIME INTERVAL
C INITIAL STORAGE - RECORDS SEQUENTIALLY ORDERED BY TIME
C FINAL STORAGE - RECORDS CONTAIN TIME SEQUENTIAL DATA FOR SINGLE
C CHANNEL
C
C A(I) = ARRAY USING AVAILABLE CORE STORAGE FOR REORDERING DATA
C NCTOT = NUMBER OF CHANNELS (PLUS 1 FOR TIME VALUE) IN DISK RECORD
C NT = NUMBER OF TIME RECORDS FOR TIME INTERVAL
C NTREC = MAX. NUMBER OF TIME RECORDS POSSIBLE IN CORE CONCURRENTLY
C NDISK = NUMBER OF TEMPORARY DISK STORAGE AREAS (MAX OF 5)
C NTREM = NUMBER OF RECORDS REMAINING TO BE READ AFTER MULTIPLES OF
C NTREC COMPLETED
C NCHAN = MAX. NUMBER OF CHANNELS ON EACH TEMPORARY DISK STORAGE
C AREA
C B(I,J) = ARRAY TO ORDER MAXIMUM OF 100 TIME DATA VALUES (SUBSCRIPT
C I) BY CHANNEL (SUBSCRIPT J)
C C(I,J) = ARRAY TO ORDER ALL TIME DATA VALUES (SUBSCRIPT I) BY
C CHANNEL FOR MAX OF 221 CHANNELS
C NCREM = NUMBER OF CHANNELS REMAINING FOR PARTIALLY-FILLED DISK
C STORAGE AREA
C NTSET = NUMBER OF DATA SETS ON EACH TEMPORARY STORAGE AREA
C (MAX OF 5)
C NCN(I) = NUMBER OF CHANNELS ON TEMP. DISK AREA I (MAX OF NCHAN)
C NTN(I) = NUMBER OF TIME VALUES PER RECORD FOR DATA SET I (MAX OF
C NTREC)
C
0001 REAL*8 DUM
0002 DIMENSION A(110500), B(100,1101), C(500,221)
0003 DIMENSION NCN(5), NTN(5)
0004 EQUIVALENCE (A(1),B(1,1),C(1,1))
0005 DEFINE FILE 14(1102,500,U,ICOUNT)
0006 FIND (14*1)
0007 CALL ASTMET(IT1A)
0008 NTREC=100
0009 NCHAN=221
C
C DETERMINE NUMBER OF TEMPORARY DISK STORAGE AREAS NEEDED (MAX OF 5)
C
0010 REWIND 8
0011 READ (8)NCTOT,NT
0012 NCTOT=NCTOT+1
0013 NDISK=NCTOT/NCHAN
0014 IF(NDISK.EQ.0)GO TO 5
0015 DO 3 I=1,NDISK
0016 NCN(I)=NCHAN
0017 3 CONTINUE
0018 5 CONTINUE
0019 NCREM=NCTOT-NDISK*NCHAN
0020 IF(NCREM.EQ.0)GO TO 7
0021 NDISK=NDISK+1
0022 NCN(NDISK)=NCREM
0023 7 CONTINUE
C
C DETERMINE NUMBER AND LENGTH OF DATA SETS
C
0024 NTSET=NT/NTREC
0025 IF(NTSET.EQ.0)GO TO 11
0026 DO 9 ND=1,NTSET
0027 NTN(ND)=NTREC
0028 9 CONTINUE
0029 11 CONTINUE
0030 NTREM=NT-NTSET*NTREC
0031 IF(NTREM.EQ.0)GO TO 13
0032 NTSET=NTSET+1
0033 NTN(NTSET)=NTREM
0034 13 CONTINUE
0035 NDSSET=NTSET
C
C READ RECORDS CONTAINING CHANNEL NAMES

```


FORTRAN IV G LEVEL 0, MOD 0

MAIN

DATE = 67216

22/08/16

PAGE 0002

```

0036      C      CALL ASTMET(IT1B)
0037      ITIME=IT1B-IT1A
0038      ITIME=(ITIME*52)/100000
0039      WRITE (6,9000)ITIME
0040      9000  FORMAT(29H0TIME - PRIOR TO REC 1 READ = ,I10)
0041      IF(NCTOT.GT.904)READ (8)DUM
0042      READ (8) DUM
0043      CALL ASTMET(IT2B)
0044      ITIME=IT2B-IT1B
0045      ITIME=(ITIME*52)/100000
0046      WRITE (6,9010)ITIME
0047      9010  FORMAT(38H0TIME - READ REC 1 AND CHANNEL NAMES = ,I10)
      C
      C      READ DISK FOR NTREC RECORDS (NTREC TIMES AND CORRESPONDING CHANNEL
      C      VALUES)
      C      WRITE DISK FOR NCTOT RECORDS OF NTREC LENGTH FOR NDISK+1 TEMPORARY
      C      DATA AREAS
      C      DISK STORAGE - T(1),C(1,1),C(2,1),...,C(NCTOT-1,1),T(2),C(1,2),...
      C      CORE STORAGE - T(1),T(2),...,T(NTREC),C(1,1),C(1,2),...,C(1,NTREC)
      C                      ,C(2,1),...,C(NCHAN,NTREC)
      C                      (B(1-100,1),B(1-100 2),...)
      C
0048      NTIC=0
0049      DO 100 ND=1,NDSET
      C
      C      READ INITIAL DISK RECORDS
      C
0050      CALL ASTMET(IT3A)
0051      NTR=NTN(ND)
0052      DO 30 NR=1,NTR
0053      READ (8)(B(NR,J),J=1,NCTOT)
      C
      C      TIME INTERVAL INCOMPLETE - REDUCE NT ACCORDINGLY
      C
0054      25  IF(NR.GT.1)GO TO 27
0055      IF(ND.EQ.1) GO TO 30
0056      IF(B(1,1).GT.B(NTN(ND-1),1)) GO TO 30
0057      NTSET=ND-1
0058      NT=NTSET*NTREC
0059      GO TO 110
0060      27  IF(B(NR,1).GT.B(NR-1,1)) GO TO 30
0061      NTN(ND)=NR-1
0062      NTSET=ND
0063      NT=NTREC*(ND-1)+NTN(ND)
0064      NTIC=1
0065      30  CONTINUE
0066      CALL ASTMET(IT3B)
0067      ITIME=IT3B-IT3A
0068      ITIME=(ITIME*52)/100000
0069      WRITE (6,9020)ITIME
0070      9020  FORMAT(30H0TIME - READ NTN(N) TIME REC = ,I10)
      C
      C      WRITE INTERMEDIATE DISK RECORDS
      C
0071      NTC=NTN(ND)
0072      NCF=0
0073      DO 60 I=1,NDISK
0074      NWD=8+I
0075      NC=NCF+1
0076      NCF=NCF+NCN(I)
0077      DO 40 K=NC,NCF
0078      WRITE (NWD)(B(L,K),L=1,NTC)
0079      40  CONTINUE
0080      60  CONTINUE
0081      CALL ASTMET(IT4B)
0082      ITIME=IT4B-IT3B
0083      ITIME=(ITIME*52)/100000
0084      WRITE (6,9030)ITIME
0085      9030  FORMAT(38H0TIME - WRITE NTSETS OF NTN(N) TIMES = ,I10)

```

FORTRAN IV G LEVEL 0, MOD 0

MAIN

DATE = 67216

22/08/16

PAGE 0003

```

0086      IF(NTIC.EQ.1)GO TO 110
0087      100 CONTINUE
0088      110 CONTINUE

C
C      READ EACH TEMPORARY DATA SET ORDER VALUES FOR EACH CHANNEL BY TIME
C      WRITE RECORD FOR EACH CHANNEL ON DISK
C      DISK STORAGE - T(1),T(2),....,T(NTREC),C(1,1),C(1,2),....,C(NCHAN,
C                      NTREC),T(NTREC+1),....,T(NT+1-NTREM),T(NT+2-NREM),
C                      ....,T(NT),C(1,NT+1-NTREM),....,C(NCHAN,NT)
C
C      CORE AND DISK STORAGE - T(1),T(2),....,T(NT),C(1,1),C(1,2),....,C(1,
C                      NT),....,C(NCTOT-1,NT)
C
0089      J=1
0090      WRITE(14*J)NCTOT,NT
0091      DO 200 I=1,NDISK
0092      NWD=8+I
0093      REWIND NWD
0094      NCH=NCN(I)
0095      L2=0
0096      CALL ASTMET(IT5A)
0097      DO 130 N=1,NTSET
0098      L1=L2+1
0099      L2=L2+NTN(N)
0100      DO 120 K=1,NCH
0101      READ (NWD)(C(L,K),L=L1,L2)
0102      120 CONTINUE
0103      130 CONTINUE
0104      CALL ASTMET(IT5B)
0105      ITIME=IT5B-IT5A
0106      ITIME=(ITIME*52)/100000
0107      WRITE (6,9040)ITIME
0108      9040 FORMAT(39H0TIME - READ NCN(I) RECORDS OF NTSETS = ,I10)
0109      DO 180 K=1,NCH
0110      J=J+1
0111      WRITE (14*J)(C(L,K),L=1,NT)
0112      180 CONTINUE
0113      CALL ASTMET(IT6B)
0114      ITIME=IT6B-IT5B
0115      ITIME=(ITIME*52)/100000
0116      WRITE (6,9050)ITIME
0117      9050 FORMAT(42H0TIME - WRITE NCN(I) RECORDS OF NT TIMES = ,I10)
0118      200 CONTINUE
0119      300 STOP
0120      END

```

```

C
C STEP 3 OF XE PLOTTING
C WRITE PLOT TAPE USING DATA STORED ON DISK
C
C   DEFINE FILE10(1102,500,U,IC)
C   COMMON /COM1/ NMPL0T, NT, LX(6,6), NSX(500), IDF(18), M
C   FIND (10*1)
C
C READ INPUT TO PLOT DATA FOR SINGLE TIME INTERVAL
C
C CALL TINPUT
C
C INITIALIZE FOR PLOTTING AND PLOT ID FRAME
C
C   IREW=1
C   ICAM=9
C   CALL PLOTID(IDF,IREW,ICAM)
C   WRITE (6,2000)(IDF(L),L=1,18)
2000 FORMAT(15H0PLOT NO.      1 ,2X,18A4)
C
C GENERATE MULTIPL0TS AND PRINT AFTER EACH PLOT IS COMPLETED
C
C CALL GENER
C
C TERMINATE PLOTTING
C
C   NMP=NMPL0T+2
C   NMFR=2+2*NMPL0T
C   WRITE (6,2020)NMP,NMFR
2020 FORMAT( 9H0PLOT NO.    ,I6 ,2X, 9HEND FRAME /30H0TOTAL NUMBER OF PLO
1T FRAMES = ,I6 )
C   CALL PLTEND(IREW)
C
C PRINT INDEX OF MULTIPL0TS AND THE MULTIPL0T TITLE
C PRINT INDEX OF CHANNELS AND CORRESPONDING PLOTS USING EACH CHANNEL
C
C ->CALL INDEX
C   STOP
C   END
C   SUBROUTINE GENER
C
C GENERATE 2 FRAMES FOR EACH MULTIPL0T
C
C   COMMON /COM1/ NMPL0T, NT, LX(6,6), NSX(500), IDF(18), M
C   INTEGER*2 ICHAN
C   REAL*8 DCHAN
C   COMMON /COM12/ NYSCL, TITLE(18,500), KGROUP(3), LY(6,6,3) ,
C   1   NCHAN(10,3), GLAB(8,10,3) , DCHAN(1100), NDCHAN,
C   2   ICHAN(32,500), SCALY(3), ISCALY(3) ,ICLASS ,NCHANT(500)
C   DO 100 M=1,NMPL0T
C
C READ MULTIPL0T INPUT
C
C CALL MINPUT
C FIND (10*ICHAN(1,M)+2)

```

```

C
C   GENERATE FRAME 1 WITH VERTICAL SCALES AND IDENTIFICATION
C
C   CALL ASTMET(IT4B)
C   CALL FRAM1
C   CALL ASTMET(ITSB)
C   TIMEA=ITSB-IT4B
C   TIMEA=TIMEA*.000052
C   WRITE (6,9040)TIMEA
9040 FORMAT(21H0TIME (SEC) - FRAM1 = ,F10.1)
C
C   GENERATE FRAME 2 WITH LABELED VECTOR PLOTS OVER TIME INTERVAL
C
C   CALL FRAM2
C   CALL ASTMET(IT6B)
C   TIMEA=IT6B-ITSB
C   TIMEA=TIMEA*.000052
C   WRITE (6,9050)TIMEA
9050 FORMAT(21H0TIME (SEC) - FRAM2 = ,F10.1)
C
C   PRINT PLOT FRAME NO. AND MULTIPLY LABEL
C
C   NFC=M+1
C   WRITE (6,2000)NFC,(TITLE(L,M),L=1,18)
2000 FORMAT( 9H0PLOT NO. ,I6,2X,18A4)
100 CONTINUE
C   RETURN
C   END
C   SUBROUTINE MINPUT
C
C   INPUT FOR A SINGLE MULTIPLY READ
C
C   NYSCL = NUMBER OF VERTICAL SCALES FOR MULTIPLY M (MAX OF 3)
C   YMIN(J) = MINIMUM VERTICAL SCALE VALUE OF SCALE J
C   YMAX(J) = MAXIMUM VERTICAL SCALE VALUE OF SCALE J
C   KGROUP(J) = NUMBER OF GROUPS USING SCALE J
C   NCHAN(K,J) = NUMBER OF CHANNELS FOR GROUP K OF SCALE J
C   (SEE NOTE ON RESTRICTIONS)
C   CHAN(IKJ) = CHANNEL NAME I FOR GROUP K OF SCALE J. THE NAMES ARE
C   PACKED CONSECUTIVELY IN STORAGE.
C   GLAB(1-4,K,J) = LINE 1 OF LABEL FOR GROUP K, SCALE J
C   GLAB(5-8,K,J) = LINE 2 OF GROUP LABEL
C   TITLE(L,M) = MULTIPLY M TITLE
C   ICHAN(IKJ,M) = INDEX OF CHANNEL CHAN(IKJ) OF MULTIPLY M IN DCHAN
C   ARRAY
C   COMMON /COM1/ NMPLY, NT, LX(6,6), NSX(500), IDF(18), M
C   INTEGER*2 ICHAN
C   REAL*8 DCHAN
C   COMMON /COM2/ NYSCL, TITLE(18,500), KGROUP(3), LY(6,6,3),
1 --- NCHAN(10,3), GLAB(8,10,3), DCHAN(1100), NDCHAN,
2 ICHAN(32,500), SCALY(3), ISCALY(3), ICLASS, NCHAN(500)
C   REAL*8 CHAN(32)
C
C   READ CARDS 5-10 FOR MULTIPLY
C
C

```

```

      NC=0
      READ (5,1030)(TITLE(L,M),L=1,18)
1030  FORMAT(18A4)
      READ (5,1000)NYSCL,ICLASS
1000  FORMAT(12I6)
C
C   READ CARDS 7-10 FOR EACH VERTICAL SCALE
C
      DO 200 J=1,NYSCL
      READ (5,1010)YMIN,YMAX,KGROUP(J)
1010  FORMAT(2=12.6,I6)
      IF(YMIN.GT.YMAX)GO TO 500
C
C   DETERMINE SCALING FACTORS AND 5-DIGIT LABELS AT TICK MARKS FOR
C   EACH VERTICAL SCALE
C
      CALL TICLAB(YMIN,YMAX,NYLOW,NDY,LY(1,1,J),0)
      SCALY(J)=1000./FLOAT(NDY)
      ISCALY(J)=19-IFIX(SCALY(J)*FLOAT(NYLOW))
      KGRP=KGROUP(J)
C
C   READ CARDS 8-10 FOR EACH GROUP
C
      DO 100 K=1,KGRP
      READ (5,1000)NCHAN(K,J)
      NCKV=NCH+1
      NC=NC+NCHAN(K,J)
      READ (5,1020)(CHAN(I),I=NCKM,NC)
1020  FORMAT(6(A8,4X))
      READ (5,1030)(GLAB(L,K,J),L=1,8)
      100 CONTINUE
      200 CONTINUE
      NCHAN(M)=NC
      DO 300 I=1,NC
      DO 250 N=1,NCHAN
      IF(DCHAN(N).EQ.CHAN(I))GO TO 275
      250 CONTINUE
      GO TO 510
      275 ICHAN(I,N)=N
      300 CONTINUE
      RETURN
C
C   ERROR
C
      500 WRITE (6,5015)N,M,J,YMIN,YMAX
      5015 FORMAT(1H0/18H FOR TIME INTERVAL ,I3,11H, MULTIPLIER ,I4, 1X, 9HAND
      1 SCALE ,I2,1X, 52HTHE MINIMUM SCALE VALUE IS GREATER THAN THE MAXI
      2MUM. /16H MINIMUM VALUE = ,F10.1 ,10X,15HMAXIMUM VALUE = ,F10.1 /
      318H CHECK YOUR INPUT. )
      STOP
      510 WRITE (6,3020)M,CHAN(1)
      3020 FORMAT(1H0/13HFOR MULTIPLIER ,I4,1X, 7HCHANNEL ,1X,A8,1X,44HIS NOT
      1INCLUDED IN THE CHANNEL DISK STORAGE. )
      STOP
      END

```

```

C      SUBROUTINE FRAM1
C
C      PLOT VERTICAL SCALE FRAME FOR MULTILOT M
C
COMMON /COM1/ NMPLT, NT, LX(6,6), NSX(500), IDF(18), M
INTEGER*2 ICHAN
REAL*8 DCHAN
COMMON /COM2/ NYSCL, TITLE(18,500), KGROUP(3), LY(6,6,3) ,
1  NCHAN(10,3), GLAB(8,10,3) , DCHAN(1100), NDCHAN,
2  ICHAN(32,500), SCALY(3), ISCALY(3) ,ICLASS ,NCHANT(500)
DIMENSION CLAB(16)
DATA DUM/' ' /
DATA CLAB/'CONFIDENTIAL RESTRICTED DATA  ATOMIC ENERGY ACT 1954  G
1GROUP 1 ' /
C      ADVANCE FRAME
C
C      CALL FRAMEV(3)
C
C      PLOT SCALE WITH LABELS AND CORRESPONDING GROUP AND CHANNEL LABELS
C
ICN=0
IG=0
IGG=0
NY=19
NX=1020-200*(NYSCL-1)
DO 100 J=1,NYSCL
C
C      GENERATE VERTICAL SCALES USING RIGHT-MOST SCALES
C
IF(J.EQ.NYSCL)GO TO 15
CALL LINEV(NX,170,NX,1019)
IF(ICLASS.EQ.0)GO TO 10
CALL LINEV(NX,NY,NX,50)
CALL LINEV(NX,90,NX,110)
GO TO 15
10 CALL LINEV(NX,NY,NX,110)
15 CONTINUE
C
C      PLACE TICK MARKS AND 5-DIGIT SCALE VALUES ON SCALE
C
C
NXD=NX-58
NYT=NY
NXT=NX-3
DO 30 IC=1,6
CALL PLOTV(NXT,NYT,32)
NXDD=NXD
DO 20 L=1,6
CALL PLOTV(NXDD,NYT,LY(L,IC,J))
NXDD=NXDD+8
20 CONTINUE
NYT=NYT+200
30 CONTINUE
C
C      GROUP AND CHANNEL LABELS
C

```

```

NEWX=NXD-138
NEWXP=NEWX+16
NEWX2=NEWX+8
NEWX3=NEWX+32
NEWX4=NEWX+48
NEWY=1000
KG=KGROUP(J)
DO 90 K=1,KG
NC=NCHAN(K,J)
CALL APRNTV(8,0,16,GLAB(1,K,J),NEWX,NEWY)
NEWY=NEWY-12
CALL APRNTV(8,0,16,GLAB(5,K,J),NEWX,NEWY)
NEWY=NEWY-12
DO 80 N=1,NC
ICN=ICN+1
IG=IG+1
IF(IG.LT.10)GO TO 40
IG=0
IGG=IGG+1
40 CONTINUE
NEWY=NEWY-12
CALL PLOTV(NEWXP,NEWY,IG)
IF(IGG.NE.0)CALL PLOTV(NEWX2,NEWY,IGG)
CALL PLOTV(NEWX3,NEWY,32)
CALL APRNTV(8,0,8,DCHAN(ICHAN(ICN,M)),NEWX4,NEWY)
50 CONTINUE
NEWY=NEWY-32
90 CONTINUE
NX=NX+200
100 CONTINUE
C
C   MULTIPLY LABEL
C
CALL APRNTV(8,0,72,TITLE(1,M),440,140)
CALL LINEV(426,110,1020,110)
CALL LINEV(426,170,1020,170)
CALL LINEV(426,110,426,170)
CALL LINEV(1020,110,1020,170)
C
C   CLASSIFIED PLOT LABEL
C
IF(ICLASS.EQ.0)GO TO 150
CALL APRNTV(8,0,64,CLAB,472,70)
CALL LINEV(448,50,1000,50)
CALL LINEV(448,90,1000,90)
CALL LINEV(448,50,448,90)
CALL LINEV(1000,50,1000,90)
150 CONTINUE
C
C   ADVANCE FRAME
C
CALL FRAMEV(3)
RETURN
END
SUBROUTINE FRAM2

```

```

C
C   PLOT FRAME WITH LABELED VECTOR PLOTS FOR EACH CHANNEL OVER TIME
C   INTERVAL
C
COMMON /COM1/ NMPL0T, NT, LX(6,6), NSX(500), IDF(18), M
INTEGER*2 ICHAN
REAL*8 DCHAN
COMMON /COM12/ NYSC1, TITLE(18,500), KGROUP(3), LY(6,6,3) ,
1  NCHAN(10,3), GLAB(8,10,3) , DCHAN(1100), NDCHAN,
2  ICHAN(32,500), SCALY(3), ISCALY(3) ,ICLASS ,NCHANT(500)
DIMENSION Y(500), NSY(500)
EQUIVALENCE (Y(1),NSY(1))

C
C   SET UP GRID
C
CALL XAXISV(1,19,1001)
CALL XAXISV(1,19,1001)
CALL YAXISV(1,19,1019)
CALL YAXISV(1,19,1019)
NY=19
NXT=1
NYT=NY
DO 20 L=1,10
DO 10 LL=1,9
NXT=NXT+10
NYT=NYT+10
CALL XAXISV(1,NYT,1001)
CALL YAXISV(NXT,19,1019)
10 CONTINUE
NXT=NXT+10
NYT=NYT+10
CALL XAXISV(1,NYT,1001)
CALL XAXISV(1,NYT,1001)
CALL YAXISV(NXT,19,1019)
CALL YAXISV(NXT,19,1019)
20 CONTINUE

C
C   ADD HORIZONTAL TICK MARKS
C
NXTH=1
NYTH=13
DO 30 IC=1,6
CALL PLOTV(NXTH,NYTH,13)
NXTH=NXTH+200
30 CONTINUE

C
C   LABEL HORIZONTAL VALUES AT TICK MARKS
C
NYT=7
NXT=4
DO 40 LL=2,6
CALL PLOTV(NXT,NYT,LX(LL,1))
NXT=NXT+8
40 CONTINUE
NXT=204

```



```

      DO 60 L=2,6
      DO 50 LL=2,6
      CALL PLUTV(NXT,NYT,LX(LL,L))
      NXT=NXT+4
50  CONTINUE
      NXT=NXT+160
      IF (L.EQ.5) NXT=NXT-24
60  CONTINUE
C
C
      CALL APINTV(1,0,4,FILE(14,*),894,7)
      CALL LINPV(884,2,936,2)
      CALL LINPV(884,3,936,12)
      CALL LINPV(884,2,884,12)
      CALL LINPV(936,2,936,12)
C
C  CHANNEL NUMBER PRINT INITIALIZATION
C
      DO 110 I=2,111
      IF (IS<(I)*.GT.20) GO TO 120
110  CONTINUE
120  IIT=NN
      IC=0
      ICC=0
C
C  PLOT ALL CHANNELS FOR MULTILOT M
C
C
C  PLOT VECTORS WITH CHANNEL LABEL
C
      I=0
      NC6=5*NCHAN(M)
      DO 200 J=1,NYSCL
      KGRP=KGRUUP(J)
      DO 190 K=1,KGRP
      NC=NCHAN(K,J)
      DO 180 L=1,NC
      I=I+1
      IFC=IIT
      IC=IC+1
      IF (IC.LT.10) GO TO 122
      IC=0
      ICC=ICC+1
122  CONTINUE
C
C  READ DATA FOR NT TIME VALUES OF CHANNEL I FROM DISK
C
      READ (10*ICHAN(I,M)+2)(Y(ICD),ICD=1,NT)
C
C  FIND LOCATION OF NEXT CHANNEL TO BE PLOTTED (ON DISK)
C
      IF (I.NE.NCHAN(M)) FIND (10*ICHAN(I+1,M)+2)
C
C  SCALE CURRENT CHANNEL VALUES AND PLOT VECTORS WITH CHANNEL LABELS
C

```

```

      YY=Y(1)
      NSY(1)=ISCALY(J)+IFIX(SCALY(J)*Y(1))
      IF(NSY(1).LT.19)GO TO 123
      IF(NSY(1).LE.1019)GO TO 124
123  IND1=1
      WRITE(6,9000)I,YY
9000  FORMAT(6H0CURVE ,13,1X,31HIS OFF-SCALE WITH DATA VALUE OF ,E12.4)
      GO TO 125
124  IND1=0
125  CONTINUE
      DO 160 NN=2,NT
      YY=Y(NN)
      NSY(NN)=ISCALY(J)+IFIX(SCALY(J)*Y(NN))
      IND2=IND1
      IF(NSY(NN).LT.19)GO TO 126
      IF(NSY(NN).GT.1019) GO TO 126
      IND1=0
      IF(IND2.EQ.0) GO TO 129
      GO TO 128
126  IND1=1
      IF(IND2.EQ.0)WRITE(6,9000)I,YY
128  CONTINUE
      IF(NN.GE.ITC)ITC=ITC+NC6
      GO TO 160
129  CONTINUE
      IF(NN.LT.ITC)GO TO 130
      ITC=ITC+NC6
      CALL PLOTV(NSX(NN),NSY(NN),IC)
      IF(ICC.EQ.0)GO TO 130
      NS=NSX(NN)-8
      CALL PLOTV(NS,NSY(NN),ICC)
130  CONTINUE
      CALL LINEV(NSX(NN-1),NSY(NN-1),NSX(NN),NSY(NN))
160  CONTINUE
      ITT=ITT+5
180  CONTINUE
190  CONTINUE
200  CONTINUE
C
      RETURN
      END
      SUBROUTINE INDEX
C
C  GENERATE INDEX OF MULTILOTS AND CORRESPONDING MULTILOT TITLE
C  GENERATE INDEX OF CHANNELS AND CORRESPONDING PLOTS
      INTEGER*2 ICHANT
      COMMON /COMIX/EP , DATE, CTITLE(14,1100), ICHANT(1100), NCTAB
      COMMON /COMI/ NMPLT, NT, LX(6,6), NSX(500), IDF(18), M
      INTEGER*2 ICHAN
      REAL*8 DCHAN
      REAL*8 EP,DATE
      COMMON /COMI2/ NYSCL, TITLE(18,500), KGROUP(3), LY(6,6,3) ,
1     NCHAN(10,3), GLAB(8,10,3), DCHAN(1100), NDCHAN ,
2     ICHAN(32,500), SCALY(3) ,ISCALY(3), ICLASS, NCHANT(500)
      DIMENSION T(9)

```

```

        IC=60
        DO 50 M=1,NMPL0T
        IF(IC.LT.60)GO TO 10
        WRITE (6,2000)EP,DATE
2000 FORMAT(1H1/1H ,14X,18HPLOT SUBJECT INDEX /1H ,14X,16HX-ENGINE TEST
        1 EP ,A8/1H ,14X, 5HDATED ,1X,A8/ 5H0CODE ,5X,10HPLOT TITLE //)
        IC=6
        10 WRITE (6,2010)TITLE(18,M),(TITLE(L,M),L=1,17)
2010 FORMAT(1H ,A4,5X,17A4)
        IC=IC+1
        50 CONTINUE
        IC=60
        DO 100 N=1,NCTAB
        IF(IC.LT.60)GO TO 60
        WRITE(6,2020)EP,DATE
2020 FORMAT(1H1/ 1H ,14X,18HPLOT CHANNEL INDEX /1H ,14X,16HX-ENGINE TES
        1T EP ,A8/1H ,14X,5HDATED ,1X,A8/ 8H0CHANNEL ,6X,12HCHANNEL NAME ,
        2 48X, 9HPLOT CODE //)
        IC=6
        60 CONTINUE
        K=0
        IK=0
        DO 80 M=1,NMPL0T
        NC=NCHANT(M)
        DO 70 J=1,NC
        IF(ICHAN(J,M).NE.ICHANT(N))GO TO 70
        K=K+1
        T(K)=TITLE(18,M)
        IF(K.LT.5)GO TO 68
        IF(IK.EQ.0)GO TO 64
        WRITE (6,2030)(T(L),L=1,5)
2030 FORMAT(1H ,73X, 5(A4,4X))
        K=0
        IC=IC+1
        GO TO 68
        64 WRITE (6,2040)DCHAN(ICHANT(N)),(CTITLE(L,N),L=1,14),(T(LL),LL=1,5)
2040 FORMAT(1H ,A8,5X,14A4,4X,5(A4,4X))
        IK=1
        IC=IC+1
        K=0
        68 CONTINUE
        70 CONTINUE
        80 CONTINUE
        IF(K.EQ.0)GO TO 100
        IF(IK.EQ.0)GO TO 90
        WRITE (6,2030)(T(L),L=1,K)
        GO TO 95
        90 WRITE (6,2040)DCHAN(ICHANT(N)),(CTITLE(L,N),L=1,14),(T(LL),LL=1,K)
        95 IC=IC+1
        100 CONTINUE
        RETURN
        END

```

BLANK

APPENDIX C

ETS-1 ENGINEERING-UNIT TAPE FORMAT

Two types of engineering-unit (EU) tapes are recorded, a long record tape and a short record tape. Each long record (996 CDC 3200 computer words) contains all the data from the corresponding frame on the gapless tape. Each short record (488 CDC 3200 computer words) contains all the data from the corresponding frame on the gapless tape, except that only one sample has been retained for each group of multisample functions. That is, for a function sampled five or ten times during one frame, only one of the five or one of the ten samples is recorded in the short record.

At the physical beginning of each EU tape is a tape header (recorded as one four-word record) of the following format in terms of 3200 computer words):

<u>Word No.</u>	<u>Contents</u>	<u>Format</u>
1	Test number	24-bit integer
2	Subsystem number	24-bit integer
3	No. of this tape	24-bit integer (consecutive, starting at 1)
4	Code	24-bit integer (two for long record tape, four for short record tape)

At the beginning of each file (and following each tape header) on the EU tape is a file header (recorded as one four-word record) of the following format (in terms of 3200 computer words):

<u>Word No.</u>	<u>Contents</u>	<u>Format</u>
1	Mode ID	See page173
2	Mode number	See page175
3	File number	24-bit integer (consecutive, starting at 1)
4	Continuation flag	24-bit integer (0 for no continuation, 1 for file continued from previous tape)

Appendix C

Each file is terminated by an end-of-file mark (followed by file header for next file). The last file on the tape is terminated by two end-of-file marks.

At the physical end of each EU tape are four special records, written immediately after the last data or header record containing the end-of-tape mark. The physical end of the tape can be recognized by sensing the end-of-tape marker or recognizing these special records. Each record consists of four CDC 3200 computer words, each word containing the four six-bit characters 4Q4Q.

All records (data, header, and end-of-tape) are recorded in the binary mode. All words in a data record (996 or 488 words) are in one of three formats: 24-bit integer, 24-bit floating point, or event format.

Those words representing events are unchanged in format compared with the corresponding words on the gapless SEL600 recording tape. Each bit in the word represents a binary event.

All other words in the record are in binarized form, as either 24-bit integers or 24-bit floating-point values. The time word and three frequency data words are integers. All other words, where not converted to EU, are in integer form and represent DDS counts; where converted to EU, they are in floating point and represent the appropriate EU value (such as °F, pressure or lb/sec).

In the case of both integers and floating-point values, negative numbers are represented in one's-complement form. In other words, in both cases, the sign of the number can be changed by complementing all 24 bits of the word. In both cases, the sign bit (left-most bit) is zero for positive numbers.

Appendix C

The three frequency data words (in integer form) represent total cycles from the start of a given tape file for the three functions being measured. The frequency counter resets at the start of a new tape file.

The 24-bit floating-point format is explained below:

Mantissa: The mantissa is a normalized fraction. That is, the assumed binary point is at the far left, and the fraction is left-adjusted against this assumed binary point, so that leading zeroes (assuming the mantissa is positive) have been dropped.

Characteristic: The characteristic is the number of places the assumed binary point would be shifted right (if the characteristic is positive) or left (if the characteristic is negative) to convert the mantissa to a fixed-point quantity.

Characteristic Bias: The characteristic is biased by a 1 in this bit position. For example, a characteristic of 0 would appear as 100000, including the bias. Further examples of characteristics, including the bias are:

<u>Biased Characteristic</u> <u>(in binary notation)</u>	<u>True Characteristic</u> <u>(in decimal notation)</u>
111111	31
100010	2
100001	1
100000	0
011110	-1
011101	-2
000000	-31

Appendix C

The only exception to this rule occurs if both the mantissa and exponent are zero; the bias bit is then also zero.

The sequence of words (CDC 3200 computer words) in a data record is as follows:

Long Record:

<u>Word No.</u>	<u>DDS Multiplexer Channel</u>
1	Range time (in tenths of seconds, starting at mid-night; i.e., time runs from 0 to 864,000)
2-78	A1004-A1080
79-156	A2003-A2080
157-273	B1004-B1120
274-390	B2003-B2119
391-590	C1001-C1040 (Each group of five consecutive word numbers contains five samples from the same multiplexer channel.)
591-780	D1002-D1020 (Each group of ten consecutive word numbers contains ten samples from the same multiplexer channel.)
781-980	E1001-E1020
981-990	D1001 (Events data)
991	A1002
992	B1002 (Events data)
993	A2002
994	B2002
995	A1003 (Frequency data)
996	B1003

Appendix C

Short Record:

<u>Word No.</u>	<u>DDS Multiplexer Channel</u>
1	Range time (in tenths of seconds, starting at mid-night; i.e., time runs from 0 to 864,000)
2-78	A1004-A1080
79-156	A2003-A2080
157-273	B1004-B1120
274-390	B2003-B2119
391-430	C1001-C1040
431-449	D1002-D1020
450-469	E1001-E1020
470-479	D1001
480	A1002 (Events data)
481	B1002
482	A2002
483	B2002
484	A1003 (Frequency data)
485	B1003
486-488	Each word contains four 6-bit blanks.

If any data (time, counts, frequency, or EU) are bad, that word is set to 24 binary ones (which, as a numerical value, is -0 in a one's-complement system). This is a unique value that will never represent good data. Bad data results from the appearance of illegal BCD characters on the original gapless tape.

The mode ID and mode number recorded in the file headers have the following description:

Mode ID: There are three mutually exclusive recording modes: continuous, multiple, and periodic. The right-hand 12 bits of the first word

Appendix C

of the file header indicate which recording mode the DDS was in for a given tape file at the time the gapless tape was recorded. If the right-most 12 bits are numbered 1 through 12 (from right to left), the meaning of the bits (when set to 1) is as follows:

<u>Bit Position</u>	<u>Meaning</u>
1	Periodic mode (1 record per second)
2	Periodic mode (1 record per 10 seconds)
3	Periodic mode (1 record per 100 seconds)
4	Always complement of bit 3
5	Not used
6	Continuous mode (1 record per 1/10 second)
7	Multiple mode (1 record)
8	Multiple mode (4 records)
9	Multiple mode (16 records)
10	Always complement of bit 9
11	Not used
12	Multiple mode (64 records)

In the continuous mode, one record is recorded every 1/10 sec, continuously, until the stop button is manually depressed. The latter causes an end-of-file to be recorded on the gapless tape (which is carried through to the EU tape), and recording stops.

In the periodic mode, one record is recorded every 1, 10, or 100 sec, continuously, until the stop button is manually depressed. The latter causes an end-of-file to be recorded on the gapless tape (which is carried through to the EU tape), and recording stops.

In the multiple mode, one record is recorded every 1/10 sec until 1, 4, 16, or 64 records have been recorded; then recording stops automatically. An end-of-file will not be recorded on the gapless tape at this time unless the stop button is manually depressed.

Appendix C

The recording mode cannot be changed unless the stop button is depressed. Recording on the gapless tape may continue, after a manual or automatic stop, by depressing the start button.

Mode Number: The mode number (in the second word of the file header) is a binary integer between 0 and 9. The source of this code is a switch setting on the DDS console, which cannot be changed unless the stop button is depressed (causing an end-of-file to be written on the gapless tape and a new file to start when recording is resumed).

There are five decade switches (settings 0 through 9) on the DDS master console, the use of which causes information to be recorded on the digital magnetic tapes of the DDS. These five switches comprise one operating-mode selector switch and four test-number selector switches.

Operating-Mode Selector Switch

This switch will be employed on all tests to record a number that will characterize the testing mode under which the data were acquired.

Test-Number Selector Switches

The first switch will be used to describe the type of test article; the second and third switches will be used to describe the sequential occurrence of tests, 00 through 99; the fourth switch will be used to describe the year.

<u>Position of Operating-Mode Switch</u>	<u>Definition</u>
0	Setup and checkout
1	Dry run
2	Pretest calibration
3	Chilldown
4	Run data

Appendix C

<u>Position of Operating-Mode Switch</u>	<u>Definition</u>
5	Hold
6	Posttest calibration
7	Cooldown
8	Posttest criticality
9	Unassigned

<u>Test Number</u>	<u>Definition</u>
(1) (2) (3) (4)	4 character positions
(1) Switch	Test article
Position	
0	Facilities test, NEPs, FEPs, etc.
1	XECF
2	XE-1
3	XE-2
4 - 9	Unassigned
(2) (3) Switch	Test number
Positions	
00 - 99	Sequential test number by year by test article
(4) Switch	
Positions	
0 - 9	Year (last digit)

Example:

Operating Mode	<u>(1)</u>	<u>(2) (3)</u>	<u>(4)</u>
4	0	15	7
(Run data)	(Facility test)	(15 this year)	(1967)

To identify data on EU tapes, an IBM card deck created at NTO will have an eight-character channel number and a word number where that channel's data are recorded within a tape record. The first seven characters of the channel

Appendix C

number are in columns 1 through 7; the eighth is in column 80. The word number is in columns 9 through 11. To identify events, an eight-character channel number will be as above, and the event number will be in columns 9 through 11. Decks to identify data and events will have the DDS subsystem number (1 or 2) in column 79.

BLANK

APPENDIX D

CONVERSION OF MODIFIED CDC 3200 FLOATING-POINT NUMBER TO THE EQUIVALENT IBM 360 FLOATING-POINT REPRESENTATION

The CDC 3200 representation of a floating-point number, as modified for output on the EU tapes, has the following format as a 24-bit representation: ABCCCCDDDDDDDDDDDDDDDDDD, where the value A indicates the number is positive if 0 and negative if 1. The value DD...D represents the significant digits of the number. BCCCCC represents the power of the base two for placement of the decimal point. The value is read from tape as a 24-bit value and is right-adjusted into a 32-bit IBM 360 binary value with eight leading zeros. The number is then left-adjusted in the 32-bit IBM 360 word by eight positions so that the leading eight zeros are shifted out and eight trailing zeros are shifted in. The IBM 360 representation, at this point, is ABCCCCDDDDDDDDDDDDDDDDDDO0000000. If A is 0, indicating a positive value, the binary value BCCCCDDDDDDDDDDDDDDDDDD is represented in true binary form. If A is 1, indicating a negative value, the value BCCCCDDDDDDDDDDDDDDDDDD is in a one's-complement format. Since the complement instructions of the IBM 360 work in two's-complement notation as contrasted with the one's-complement arithmetic of the CDC 3200, a recomplementing of the value on the 360 results in a magnitude one bit greater in the twenty-fourth position than the true value. For this reason, a value of one is subtracted from the twenty-fourth position of the recomplemented number. A flag is set that indicates the value was originally negative, and the computation from this point on proceeds as though the number were positive from the beginning. Unlike the CDC 3200, a negative floating-point value on the IBM 360 is not represented in two's-complement form. Therefore, if the value was negative, a negative sign must be appended after the conversion is complete.

At this point, the absolute value of the CDC modified floating-point number has been obtained left-adjusted in an IBM 32-bit register.

Appendix D

To repeat, the DD...D portion of the representation represents the significant digits of the number. Since there are only 17 bits (binary position) provided for this significance, the largest number of significant digits that can be expected is five; i.e., $2^{17} - 1 = 131,071$. Although base-10 numbers from 100,000 to 131,071 have six significant digits, base-10 numbers from 131,072 to 999,999 cannot be represented with only 17 bit positions. Hence, five is considered to be the largest number of significant digits that can be expected. The DD...D is a binary fraction that has the binary functional value DD...D, where the D's are either 0 or 1 in binary notation, although the first D must always be a 1 because all floating-point numbers are represented in a normalized form; i.e., the first D is a 1 when positive and is 1 after conversion to an absolute value when the value was originally negative.

The floating-point number, in essence, is actually a representation of a value in scientific (engineering) notation; e.g., $.36512 \times 10^3$. Since binary values are being dealt with, however, the number is now equal to $.DD...D \times 2^y$, where y is yet to be determined. To further complicate matters, the characteristic exponent in the IBM 360 is the power of a base 16 as contrasted with a base 2 in the CDC 3200. This means a number $.F...F \times 16^x$ is required in the IBM 360, which is equivalent to $.DD...D \times 2^y$ in the CDC 3200. Since the IBM 360 word has 32 bit positions, 24 bit positions are available for the binary fraction. The significance is determined by the 5-digit maximum of the argument.

The next consideration is determination of the representation for y (the exponent to the base 2 in the CDC 3200). The BCCCCC part of the absolute representation of the number determines the value of y. Since six binary bits are used to represent this value, the range of values it may cover is from 0 to $2^6 - 1$, or 63 in decimal form. Since the number contains only one sign in the first bit position, there is no obvious sign for the exponent y. Therefore, a method called excess-32 notation is used, 32 being in the middle of the range of BCCCCC (0 to 63). If the value of BCCCCC is equivalent to

Appendix D

31 or 32 decimal, the value of the exponent y is 0; i.e., the binary value of the number is $.DD...D \times 2^0$ or is $.DDD..D$ in binary. If the value of BCCCC equals 33 (one more than 32) the value of y is 1; or, if it equals 40 (8 more than 32), the value of y is 8 and the value of the number is $.DD...D \times 2^8$, where the D's are in binary and the base 2 and exponent 8 are decimal representations.

If BCCCC is 30 in decimal (one less than 31), the value of y is -1. If BCCCC is 23 in decimal (8 less than 31), the value of y is -8, yielding an equivalent value of $.DD...D \times 2^{-8}$. The exponent y indicates the positioning of the binary point to get the true value of the floating-point number; i.e., if $y = 3$, the value in binary of the number in question is $DDD.DD...D$. If $y = -3$, the value would be $.000DD...D$. The exponent y , therefore, determines the direction and extent of the shift of the binary point to represent the value in true binary. On the other hand, the exponent x in the IBM 360 floating-point representation indicates a hexadecimal shift (four bits at a time). For every multiple of 4 in the value of y , a value of 1 is needed for x ; i.e., if $y = +16$, $x = +4$; or, if $y = -12$, $x = -3$. Since y is obviously not a multiple of 4 in most cases, it must be made so. First the exponent (base 2) BCCCC is compared with the value 32 decimal. If it is less than this value, the exponent is increased by one to compensate for the fact that both 32 and 31 yield a y value of 0. After this addition of 1, a value of 31 yields a y value of -1, 30 yields a y value of -2, etc.

Next, the value BCCCC is divided by 4 and the remainder is checked. If it is now zero; i.e., if BCCCC (and thus y) is not a multiple of 4, the integral position of the quotient (the hexadecimal exponent required) is increased by 1. The assumption here (an invalid one) is that the value BCCCC was equivalent to a value of 4 minus the remainder, which would then have yielded the increased quotient and no remainder. To compensate for this invalid assumption, the fractional portion of the CDC 3200 number, $DD...D$, is shifted right a number of places equal to 4 minus the remainder. Assume the

Appendix D

value of 4 minus the remainder was equal to 1. Had the dividend been 3 greater (greater by a power of 2^3), the quotient would have been one greater with no remainder, which is what is needed; i.e., an exact multiple of 4. By shifting a binary fraction right one position, the number is halved, just as shifting a decimal fraction right one position results in a value equal to one-tenth of the original value. In this example, the shift is made 3 places to the right such that the fraction is reduced by a factor of 2^3 to compensate for the increase in the value of the integral quotient by 1. At this point, the fraction is in the form required by the IBM 360 since, in order to be in a normalized form, at least one of the first four bits of the fraction must be nonzero. The division by 4 of the CDC (excess-32) characteristic results in an excess-8 hexadecimal result. In a manner similar to that for the CDC 3200, the IBM 360 operates in an excess-64 hexadecimal characteristic. Since there is already an excess-8 characteristic, 56 (decimal) is added to this value to form the excess-64 hexadecimal characteristic. By positioning the two computed values with the characteristic in bit positions 2 through 8 and the fraction in positions 9 through 32, only the sign need yet be determined. If the sign flag was set to indicate the value was initially negative, position 1 of the result is set to 1, which indicates a negative value in the IBM 360. Otherwise the value is left as 0.

There is no loss of significance in the conversion.

RECEIVED

1967 OCT 23 PM 3:12

SNPO-C
CLEVELAND, OHIO

AEROJET-GENERAL CORPORATION